Pattern Recognition 83 (2018) 52–63

Contents lists available at ScienceDirect

Pattern Recognition

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

Clustering via binary embedding

Manuele Bicego[a,*], Mário A.T. Figueiredo[b]

[a] Computer Science Department, University of Verona, Strada Le Grazie 15, Ca' Vignal 2, Verona, Italy
[b] Instituto de Telecomunicações, and Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

ARTICLE INFO

Article history:
Received 24 August 2017
Revised 30 April 2018
Accepted 13 May 2018
Available online 14 May 2018

Keywords:
Clustering
Binary embedding
Finite mixture models
Biclustering
One-class classification

ABSTRACT

In this paper, we present a novel clustering scheme based on binary embeddings, which provides compact and informative binary representations of high-dimensional objects. The binary representations are obtained with a collection of one-class classifiers learned from (pseudo) randomly selected points in the dataset. To cluster the binary representations, we consider two approaches: a mixture of Bernoulli distributions and a recent biclustering approach called CRAFT. The empirical evaluation in comparison with both classic and recent clustering methods, based on 12 different datasets, provides encouraging results. The main feature of the proposed method is that it is agnostic to the shape of the clusters.

1. Introduction

Data clustering is a classical pattern recognition problem where the goal is to organize a collection of objects into groups, or clusters, so that objects in the same cluster are more similar among themselves than to objects belonging to other clusters [1–3]. Many different techniques have been proposed to tackle this problem, ranging from the famous *K-means*, proposed more than six decades ago [1], to more recent and sophisticated approaches, such as spectral clustering [4,5], finite mixtures [6], kernel-based methods [7,8], and many others [9–11]. All those techniques address clustering through different problem formulations, representations, criteria, and algorithms; typically, they rely (explicitly or implicitly) on assumptions or simplifications: for example, standard K-means has a built-in assumption that the clusters are expected to be spherical in the feature space [1–3].

In this paper, we propose a new clustering method based on a *binary embedding* of the objects to be clustered, i.e., where each object is represented by a binary string/vector. In binary embedding, a well-studied topic in data processing, the goal is to represent objects by *binary vectors*: this representation/embedding should be chosen so that, in the resulting binary space, the neighborhood relationships among the original objects is as well preserved as possible. Most of the times, the binary embedding is implemented via a projection followed by a binarization operation (often a sign function): in that case, research is mainly focused on the definition of the most suitable projections. Different solutions have been proposed, ranging from random projections [12,13] to more complex functions optimizing sophisticated criteria, such as reconstruction error, data dissimilarity, rankings, and others [14–17]. Typically, the goal of keeping the binary representation as compact as possible is also present [18,19].

Binary embeddings have been employed in many different scenarios, such as classification, retrieval, indexing, data storing, and others; as reported above, in most of the cases, the main goal is to embed high-dimensional objects in *lower dimensional* binary representations. In this paper, on the contrary, we devise a binary embedding scheme for clustering, where objects are embedded in a (typically) *higher-dimensional* binary space, such that data clusters are more easily discovered. More precisely, the proposed approach defines the embedding using *a set of classifiers* in the original feature space. The intuition is that points sharing the same class with respect to a large collection of classifiers are naturally expected to belong to the same cluster.

Clearly, a central aspect of the proposed approach is the definition of a proper set of flexible and meaningful classifiers. Our proposal is to use a collection of one-class classifiers [20–22], a particular type of classifiers – typically used for outlier detection – that can be trained using only examples from the *positive* class (for example, a Gaussian with a threshold, a sphere, an ellipse, or a one-class support vector machine [21]). In our approach, each one-class classifier is learned from a small subset of points from the dataset. These trained models, herein referred to as *embedding models*, are then used to derive the binary embedding as follows: a given point x is represented by a binary vector where the jth entry is 1 if x is classified in the positive class by the jth one-class classifier, and 0

* Corresponding author.
E-mail address: manuele.bicego@univr.it (M. Bicego).

https://doi.org/10.1016/j.patcog.2018.05.011
0031-3203/

otherwise. A set of embedding models may contain different families of classifiers (e.g., Gaussian *and* one-class SVM) and define different regions (see Fig. 3, for an example); this aspect, together with the fact that the points used to define the classifiers can be sampled with many different strategies (e.g. nearby, far away, near the centroid, near the boundary, nearly aligned) makes the whole scheme very flexible.

After the embeddings of all the objects in a dataset are obtained, clustering is performed on these binary signatures of the objects. To be as general as possible, and to reduce the number of assumptions to a minimum, the first approach that we propose to cluster the binary representations is by learning a mixture of Bernoulli distributions, where each component models the region membership pattern that characterizes each cluster.

In some cases, several components of the binary representation may be irrelevant for the clustering task, namely because the corresponding classifiers do not contribute to distinguish the clusters. In such cases, clustering the complete binary signatures may not lead to good results, since clustering criteria look for a coherent behaviour of objects with respect to *all* the classifiers. In these cases, it seems more reasonable to look for subsets (clusters) of objects that exhibit a coherent behaviour in a *subsets* of classifiers (hopefully, as large as possible). To handle this task, we propose resorting to *biclustering* schemes [23], where the aim is to discover groups of objects that behave coherently in groups of features. In our context, even if we have to pay for the higher computational complexity, we can potentially discover groups of objects with similar memberships pattern with respect to a subset of classifiers; this is equivalent to performing simultaneous clustering and feature selection, a long-standing problem in the clustering literature [24]. In particular, here we adopt the very recent CRAFT (ClusteR-specific Assorted Feature selecTion) algorithm [25].[1]

The proposed approach has been evaluated on a collection of classical benchmark datasets, analyzing the effect of several choices, namely the number of embedding models, the sampling strategy, and the final clustering technique. The obtained results suggest that the proposed scheme represents a viable alternative to classical as well to advanced clustering algorithms.

2. Related work

Although some of the ideas and tools used in the proposed approach can be found in other techniques, here they are combined in a novel way to obtain a new general-purpose and highly flexible clustering framework. In this section, we summarize some of this related work.

In the so-called *generalized dissimilarity-based representation* [26–28], each object is represented by a vector of dissimilarities/similarities with respect to a set of models (e.g., lines [28], hidden Markov models [27], one-class SVM [26]). This class of approaches generalizes the *dissimilarity-based representation paradigm* [29], which claims that an effective representation of a given object can be obtained by looking at its dissimilarities with respect to other objects (prototypes); the effectiveness of this approach (and several variants thereof) has been shown in several contexts [29]. Our approach is related with this line of reasoning, in that it represents each object by a binary vector produced by a set of trained models. However, we simultaneously employ many different types of models (rather than a single type). Moreover, and somewhat surprisingly, dissimilarity-based representations have been hardly exploited for clustering.

[1] The method has been presented as a clustering approach which makes cluster specific feature selection; however it can easily be seen as an approach to determine exclusive-rows biclusters (following the notation in Fig. 4 of [23], and assuming that every row represents the embedding of a single object).

Within the same line of reasoning, the so-called *preference analysis* (PA) framework [30–32] consists of a class of methods specifically designed to face the *multiple structure recovery* (MSR) problem in computer vision. MSR aims at identifying multiple models (such as lines) from noisy data, which may also contain outliers; the kind of model (line, plane, circle, etc...) has to be pre-specified. Typically, MSR is addressed by extracting some tentative structures (obtained by random sampling), and estimating how they fit the given points; the result is a (binary) matrix where an entry (i, j) indicates if the jth model (structure) "explains" the ith point. Instead of the classical *consensus analysis*, which looks for the models that explain most of the points (i.e., methods based on the Hough transform or RANSAC [33,34]), PA reverses that viewpoint, by analyzing how the points are explained by the different models [30–32]. In particular, similarly to our approach, each point is represented by a signature which indicates its fit to the pool of tentative structures; the MSR problem is then solved by clustering these signatures. Even if sharing a similar pipeline, there are some essential differences. Whereas PA crucially exploits the basic assumption that each cluster has the same form as the model used for the embedding, in our approach the clusters do not have a pre-specified form. Moreover, our scheme is more general, relying on a library of models and dealing with general input data. Finally, to obtain the binary signatures, PA requires a threshold, which is used to define if a point is "well explained", or not, by a given model. In our proposal, in contrast, this is automatically determined by the learned classifiers.

Our method is also related with approaches that use projections to perform an embedding of objects in a feature space, where clustering is then carried out. Some methods, close in spirit to the binary embedding for supervised solutions, employ *random* projections to define the embedding space [35–40]. All those works, however, employ only projections, and are specifically designed to reduce the high dimensionality of the problem.

Finally, some weaker links can be also established with clustering ensembles [41], since we are in some sense combining different models which can potentially characterize clusters, or even with discriminative clustering [42], since we propose to obtain the binary signature by separating group of points.

Summarizing, the proposed approach exploits and merges in a unique fashion tools and ideas present in different related areas (binary compression, dissimilarity-based representation, preference analysis); the proposed framework thus represents a new general-purpose and flexible clustering approach, which is agnostic to the shape of the clusters.

3. The proposed approach

This section begins by defining the form of our binary embeddings, which uses a collection of one-class classifiers trained on small random subsets of the dataset to be clustered. Then, we discuss different strategies to obtain these random subsets and their motivations. Finally, we present the proposed binary clustering algorithms. The whole scheme is then summarized in Fig. 1. This section also contains some observations on the dimensionality issue and a toy example used to highlight and illustrate the different components of the proposed method.

3.1. Binary embeddings

The set of N objects to be clustered is denoted as $X = \{x_1, \ldots, x_N\}$, where each $x_i \in \mathcal{X}$. We consider some dissimilarity function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and in general do not assume anything else about this function (e.g., that it is a metric), unless otherwise explicitly indicated. If each object is characterized by a vector of p real-valued features, we overload the notation to let $x_i \in \mathbb{R}^p$ denote

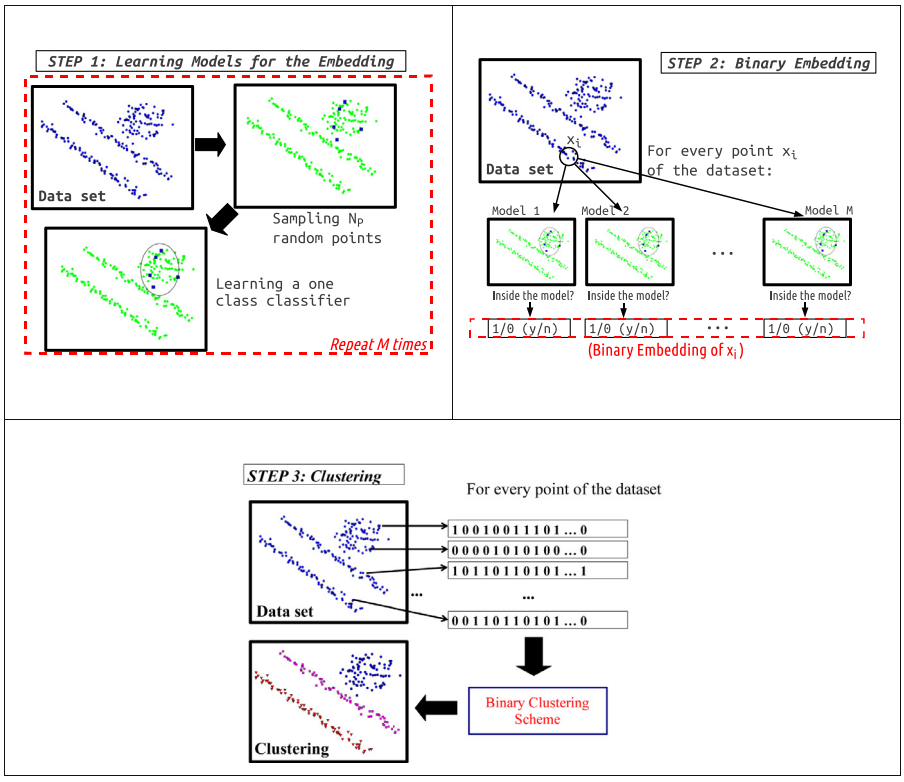

Fig. 1. Summarizing scheme of the proposed approach.

the vector representing the ith object; in this case, d will denote the Euclidean distance in $\mathbb{R}^p$, that is, $d(x_i, x_j) = \|x_i - x_j\|$.

A binary embedding of the dataset X is a mapping E: $X \to \{0, 1\}^M$, in which each element of X is mapped into a binary $M-$vector. This mapping does not have to be injective, thus it may not be invertible. The binary embedding can be written as

$$E(x_i) = \big(e_1(x_i), \ldots, e_M(x_i)\big), \tag{1}$$

where each $e_j : \mathcal{X} \to \{0, 1\}$ is a binary function. In this paper, each of these binary functions e_j is a one-class classifier [21], learned from a small random subset of X, denoted as $P_j \subset X$ (below we will discuss different strategies for obtaining these random subsets).

3.2. Sampling subsets of points

As mentioned above, each of the M one-class classifiers e_j is learned from a set P_j containing N_P points sampled from the dataset X without replacement. Typically, N_P is a small number, but it should be no less than the minimum number of points needed to learn the adopted type of one-class classifier. The random sampling can be driven by different strategies, such as giving high probability to points that are distant from previously sampled points (as is done in the K-means++ initialization method for K-means [43]) or the opposite, to favor compact subsets. In our approach, we consider the following two sampling strategies.

- *Compact sampling*: in this scheme (similar to that used in [31]), the points are preferably sampled in a neighborhood of the feature space. To this end, the first point is randomly chosen, whereas the remaining $N_P - 1$ points are chosen so that nearby points have higher probabilities. In practice, given the first point x_1 (randomly picked), the set P_i of points is incrementally obtained by sampling points from the following distribution

$$\mathbb{P}(x_n) = \frac{1}{Z} \begin{cases} \exp\big(-d(x_n, x_1)/\sigma\big) & \text{if } x_n \notin P_j \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where Z is a normalization constant and σ controls the tightness of the distribution, thus the size of effective neighborhood (see Fig. 3 (a) and (b)). Notice that for this sampling scheme, nothing needs to be assumed about the dissimilarity measure d.

- *Elongated sampling:* this scheme aims at extracting points forming an elongated set (see Fig. 3 (c), (d) and (e)); here, we are assuming that each $x_i \in \mathbb{R}^p$ and that d is the Euclidean distance. In this case, we first randomly sample the first two points x_1 and x_2, with the remaining points being sampled by giving higher probability to points that are near the line connecting x_1 and x_2. This is done by using again Eq. (2), with the distance $d(x_j, x_1)$ replaced with $d_l(x_j, \text{line}(x_1, x_2))$, which denotes the distance of point x_j to the line connecting the two points x_1, x_2. In fact, this distance can be efficiently computed by using only distances between points, which allows relaxing the assumption that d is the Euclidean distance and requiring only that it is a metric [28].

At the end of the complete sampling procedure, we obtain a collection of M subsets of X, that is, $\{P_1 \subset X, \ldots, P_M \subset X\}$. Notice that these subsets do not have to form a partition of X (in general, they do not), since they are not necessarily disjoint and their union may not be equal to X.

3.3. Obtaining the binary embedding

Each component $e_j : \mathcal{X} \to \{0, 1\}$ of the binary embedding function is a one-class classifier learned from the corresponding subset P_j. Recall that one-class classifiers are a particular type of classifier that can be trained using only positive examples [20,21].

Many one-class classifiers have been proposed, ranging from simple Gaussian models, up to support vector domain descriptors, also known as *one-class support vector machines* (OC-SVM) [44,45]. Typically, to define the decision boundary, a fraction is provided, indicating the percentage of training points that should belong to the region defining the positive class. This can be directly used in the method (like the parameter ν in the OC-SVM), or can be used to estimate a threshold. In our approach, we avoid having to set this parameter by enforcing the model to be the smallest one that classifies *all* the points in P_j as positive; for example, in the case of a sphere, the positive region is the minimum sphere containing all points in P_j.

In our method, we are given M subsets of points, $P_1, \ldots, P_M$, to train M one-class models. Even if each model is trained on a different set of points randomly sampled from the dataset, some correlation between the models can be present, since it is possible that the sets of random points overlap. A possible alternative, not investigated here, would be to sample "non-overlapping" sets, to force the subsets $P_1, \ldots, P_M$ to be disjoint. This could reduce the correlation between models, but it would may limit the number of possible models, thus reducing the final possible dimension of the binary embedding space.

Given the M classifiers, each object x_i is represented by an M-dimensional binary vector $E(x_i) = \big(e_1(x_i), \ldots, e_M(x_i)\big)$. By stacking the binary embeddings of all points, we obtain an $N \times M$ binary matrix, which represents the embedding of the whole dataset.

3.4. Clustering the binary embeddings

Given the binary matrix that results from the embedding, we perform clustering using two schemes: (a) we investigate a simple strategy, which makes very few assumptions, modelling the columns of the matrix as being generated by a Bernoulli mixture; (b) we also consider a more sophisticated scheme, the CRAFT algorithm [25], which takes into account the fact that a cluster represents a group of objects which behave coherently with respect to *a subset* of features (classifiers, in this case). Next, we review in detail the Bernoulli mixture model, whereas for CRAFT we provide only a brief summary, redirecting the interest readers to the original publication [25].

3.4.1. The Bernoulli mixture

In this section we introduce the Bernoulli Mixture model,[2] a simple probabilistic model usable to cluster binary data [46], recently extended also to the biclustering case [47,48].

The goal of this model is to cluster the N rows of the $N \times M$ matrix $\mathbf{B}$, where $B_{i,j} = e_j(x_i)$ is the jth bit of the binary embedding of x_i. We denote the ith row of $\mathbf{B}$ as $\mathbf{b}_i = E(x_i) = \big(e_1(x_i), \ldots, e_M(x_i)\big)$.

Recall that a Bernoulli mixture with K components is expressed by the following probability mass function (for $b \in \{0, 1\}$)

$$\mathbb{P}(b) = \sum_{k=1}^{K} \alpha_k \, \theta_k^b (1 - \theta_k)^{(1-b)}, \tag{3}$$

where α_k is the probability of the kth component and θ_k is the Bernoulli parameter of the kth mixture component.

Our generative model for the elements of matrix $\mathbf{B}$ is as follows: each row, say i, chooses one of K components with probabilities α_1, ... ,α_K (naturally, $\alpha_k \geq 0$ and $\sum_k \alpha_k = 1$); given the chosen component, say k, the jth element of the ith row is sampled from a Bernoulli distribution with parameter θ_{jk}, and all the elements in a row are mutually independent, conditioned on the chosen mixture component. Finally, the rows are mutually independent. Formally, this corresponds to the following joint probability function:

$$\mathbb{P}(\mathbf{B}) = \prod_{i=1}^{N} \sum_{k=1}^{K} \alpha_k \prod_{j=1}^{M} \theta_{jk}^{b_{ij}} (1 - \theta_{jk})^{(1-b_{ij})}. \tag{4}$$

Using this model to cluster the rows of $\mathbf{B}$ corresponds to obtaining estimates of its parameters, $\hat{\theta}_{jk}$ and $\hat{\alpha}_k$, for $j = 1, \ldots, M$ and $k = 1, \ldots, K$, and then assigning each row to the component with the highest posterior probability; that is, letting $z_i \in \{1, \ldots, K\}$ be the cluster label of the ith row,

$$\hat{z}_i = \arg \max_{k \in \{1,\ldots,K\}} \hat{\alpha}_k \prod_{j=1}^{M} \hat{\theta}_{jk}^{b_{ij}} (1 - \hat{\theta}_{jk})^{(1-b_{ij})}. \tag{5}$$

To estimate the model parameters, we use an *expectation-maximization* (EM) algorithm [49,50], where the missing variables are obviously the cluster labels z_i, which we represent (as is standard when deriving EM algorithms for mixture models) using binary indicators: $y_{ik} \in \{0, 1\}$, with $y_{ik} = 1$ if and only if $z_i = k$. We denote the collection of all these indicator variables as $\mathbf{Y}$. Using Υ to denote the complete set of parameters, $\Upsilon = \{\alpha_k, \theta_{jk}, \text{for } k = 1, \ldots, K, \, j = 1, \ldots, M\}$, the complete log-likelihood $\log \mathbb{P}(\mathbf{B}, \mathbf{Y}|\Upsilon)$ is given by

$$\log \mathbb{P}(\mathbf{B}, \mathbf{Y}|\Upsilon) = \log \mathbb{P}(\mathbf{B}|\mathbf{Y}, \Upsilon) + \log \mathbb{P}(\mathbf{Y}|\Upsilon) \tag{6}$$

where

$$\log \mathbb{P}(\mathbf{Y}|\Upsilon) = \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log \alpha_k \tag{7}$$

and

$$\log \mathbb{P}(\mathbf{B}|\mathbf{Y}, \Upsilon) = \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log \Big(\prod_{j=1}^{M} \theta_{jk}^{b_{ij}} (1 - \theta_{jk})^{1-b_{ij}} \Big) \tag{8}$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \sum_{j=1}^{M} \Big(b_{ij} \log \theta_{jk} + (1 - b_{ij}) \log(1 - \theta_{jk}) \Big). \tag{9}$$

The linearity of the complete log-likelihood with respect to the hidden variables y_{ik} implies that the E-step of the EM algorithm corresponds to computing the conditional expectation of these hidden variables, which are then plugged back into complete log-likelihood, yielding its conditional expectation. With $\hat{\Upsilon}$ denoting the current parameter estimates, the EM algorithm iterates between two steps:

- E-step: the conditional expectation of the hidden variables $\mathbf{Y}$ is computed, given the observed ones $\mathbf{B}$, and the current estimate of the parameters $\hat{\Upsilon}$. For our model, this corresponds to

$$w_{ik} = \mathbb{P}(y_{ik} = 1|\mathbf{B}, \hat{\Upsilon}) = \frac{\hat{\alpha}_k \prod_{j=1}^{M} \hat{\theta}_{jk}^{b_{ij}} (1 - \hat{\theta}_{jk})^{1-b_{ij}}}{\sum_{l=1}^{K} \hat{\alpha}_l \prod_{j=1}^{M} \hat{\theta}_{jl}^{b_{ij}} (1 - \hat{\theta}_{jl})^{1-b_{ij}}}. \tag{10}$$

  Let $\mathbf{W}$ be the collection of all the w_{ik} variables, such that $\mathbf{W} = \mathbb{E}[\mathbf{Y}|\mathbf{B}, \hat{\Upsilon}]$.
- M-step: the parameter estimates are updated by maximizing conditional expectation of the complete log likelihood defined in Eq. (6), that is,

$$\hat{\Upsilon} \leftarrow \arg \max_{\Upsilon} \Big(\log \mathbb{P}(\mathbf{B}|\mathbf{W}, \Upsilon) + \log \mathbb{P}(\mathbf{W}|\Upsilon) \Big) \tag{11}$$

  After some simple manipulations, we obtain the following simple update expressions:

$$\hat{\alpha}_k \leftarrow \frac{1}{N} \sum_{i=1}^{N} w_{ik} \tag{12}$$

$$\hat{\theta}_{jk} \leftarrow \frac{\sum_{i=1}^{N} w_{ik} b_{ij}}{\sum_{i=1}^{N} w_{ik}} \tag{13}$$

[2] A similar model was introduced in [6], with the name *latent class model*.

The EM algorithm starts from an initial estimate $\hat{\Upsilon} = \Upsilon^0$ and iterates between the two steps until some convergence criterion is met; typically, the relative change in log-likelihood being less than some threshold. As in many clustering algorithms [51], a good initialization is crucial to get a good model estimate: the experimental section contains the details on how we faced this problem in our experiments.

3.4.2. The CRAFT approach

The CRAFT algorithm (ClusteR-specific Assorted Feature selecTion [25]) is a very recent probabilistic approach for clustering both numerical and categorical data, which aims at selecting, for each cluster, the best set of features. The main intuition behind CRAFT is that objects belonging to a cluster should agree on the features selected for that cluster, this being similar in spirit to biclustering [23]. In the binary case we are considering, this method corresponds to the maximization of the cluster entropies over subsets of features; by deriving an asymptotic approximation [52,53] for the joint log likelihood of observed data, cluster indicators, cluster means, and feature means, the authors derive an elegant K-means style algorithm. In particular, the approach iterates between three steps: (i) computing the distances from cluster centers, using the features selected for each cluster, (ii) assigning points to clusters, (iii) recomputing cluster centers and estimating the appropriate features for each cluster. For further details please refer to [25].

3.5. The dimensionality of the binary embedding

As described above, the proposed approach builds the embedding space by exploiting a set of one-class classifiers, determined on the basis of randomly selected subsets of points. Naturally, the obtained clusters may depend on the randomly chosen points: one way to reduce the influence of this random fluctuation is to simply increase the number of random subsets (i.e., the dimensionality of the embedding). This, however, can be beneficial only up to a certain point, since too many features may cause difficulties to the binary clustering algorithms and imply an increasing computational cost. In fact, the increased computational cost and the increased number of possible irrelevant features may lead to poor results in the final clustering.

In order to address this problem, we propose a strategy with two steps: (i) obtain several different clustering solutions, each one starting from a "not too high" number of classifiers; (ii) select the clustering from the pool of solutions (either by choosing the best one or by fusing them via an ensemble clustering method [41]). With this strategy, we can maintain the embedding space with a reasonable dimension, while being robust to the possible random fluctuations due to the sampling process. We implemented two variants of this idea, which we call *Sel* (Selection) and *Ens* (Ensemble), respectively:

1. Sel (Selection): given L clusterings, we select the best one, according to the entropy criterion proposed in [54], which is specifically designed for binary data. In particular, for the lth clustering $\mathcal{C}^\ell = \{C_1^\ell, \ldots, C_K^\ell\}$, we evaluate the following criterion:

$$O(\mathcal{C}^\ell) = \frac{1}{M}\left(H(\mathbf{B}) - \frac{1}{N}\sum_{k=1}^{K} n_k H(C_k^\ell)\right), \tag{14}$$

   where $n_k = |C_k^\ell|$ is the number of points in cluster C_k^ℓ, $H(\mathbf{B})$ is the entropy of the input binary data, and $\frac{1}{N}\sum_{k=1}^{K} n_k H(C_k^\ell)$ is the entropy of the partition, i.e. the weighted sum of each cluster's entropy (for more details, see [54]). The chosen clustering $\hat{\mathcal{C}}$ is the one maximizing (14).

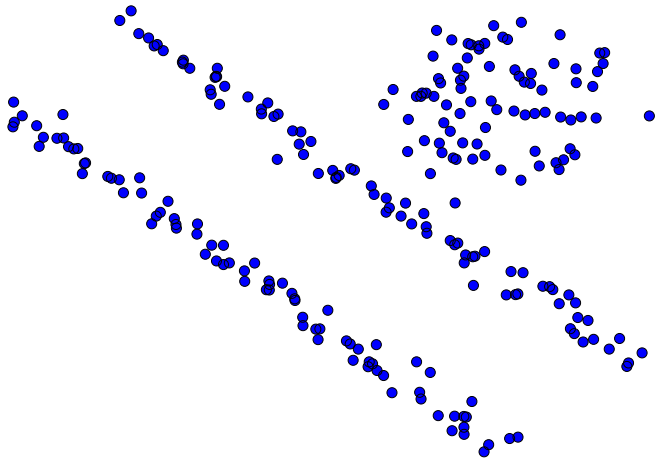

Fig. 2. Toy example.

2. Ens (Ensemble): here, instead of selecting the best clustering, we aggregate the L clusterings by using an ensemble clustering approach. There exist different approaches for ensemble clustering [55] (and also for biclustering [56]); in particular, we adopt the *evidence accumulation clustering* (EAC) scheme [41]. In that scheme, the idea is to assess the similarity between two objects via the number of times they are placed in the same cluster, in a collection of different clusterings. The rationale is that objects which belong to a "real" cluster are very likely to be assigned to the same cluster in different partitions. Following [41], we derive the final clustering using a single-linkage hierarchical clustering approach.

3.6. Computational complexity

The proposed approach represents a general-purpose approach composed by different blocks: their implementation strongly influences the complexity of the whole scheme, which is therefore difficult to quantify. In general, we observe that the heaviest step is building the one-class classifiers underlying the embedding: we have to train M different one-class classifiers, which may be costly if we have large M or if we choose complex models (such as OC-SVM). In our experiments, we used a moderately small number of models (100) and the 1-nearest-neighbor (1-NN) one-class model, which is very fast (it only computes distances from the training points) yet very accurate. The one-class classifiers are trained on the sampled points, which can be really few (we used 10 or 15 points in our experiments). Once the models are trained, the embedding is typically very fast (e.g., comparison with a threshold). The number of models also influences the complexity of the binary representations: large M results in high dimensional embeddings, and this can have an influence on the computational complexity of the binary clustering algorithm; in our case. the Bernoulli Mixture is definitely faster than the more complicated CRAFT.

3.7. Toy illustrative example

This section uses a synthetic toy example to visually highlight some of the different components of the proposed approach. The data, shown in Fig. 2, is bidimensional and has three clusters with different shapes: two are elongated and one is compact. We expect classical techniques to have difficulties in clustering this dataset, due to the specific underlying assumptions on which they rely. Fig. 3 shows the results of the different steps of the proposed method when applied to this dataset: the sampling of the points (first column), the determination of the one-class classifiers, and

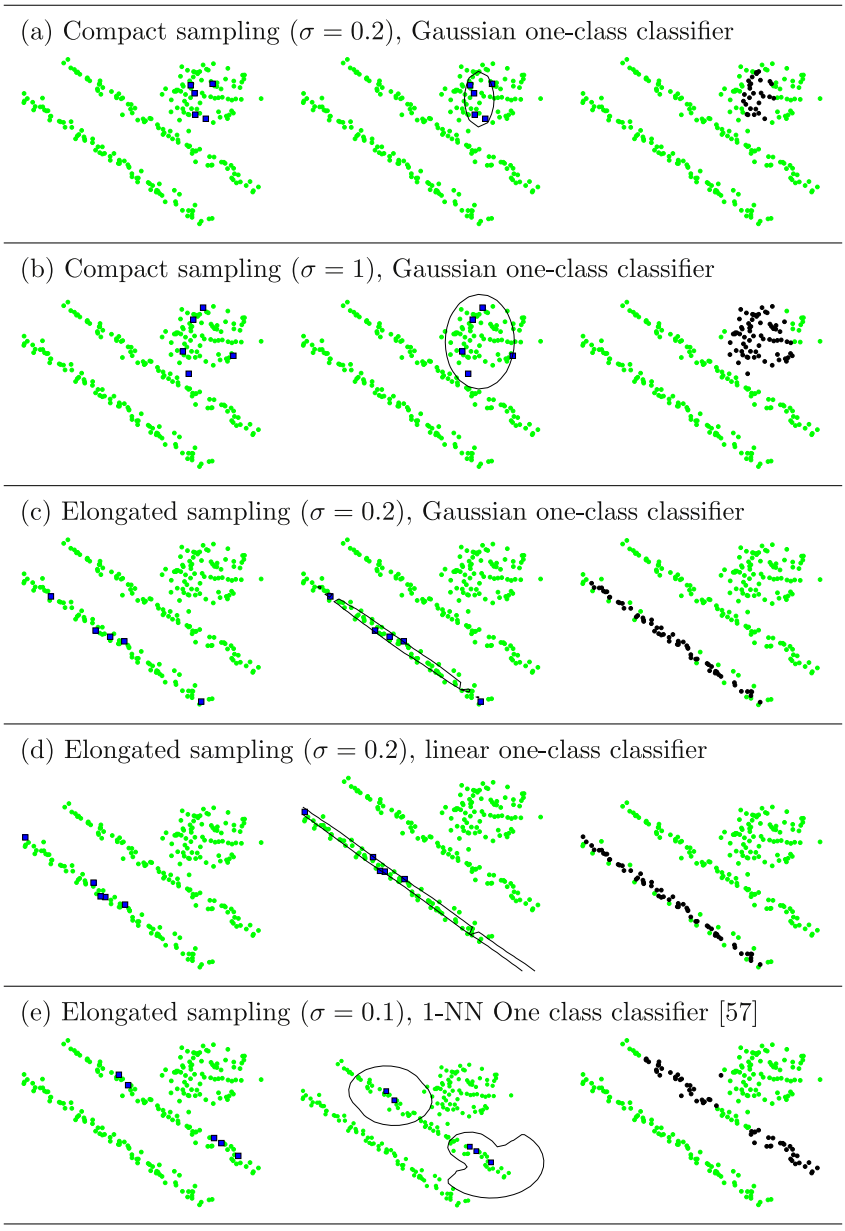

Fig. 3. (Best viewed in color) Toy example: some possible choices for binary embedding. First column: the sampled points (blue squares); Second Column: the boundary of the region; Third column: filled black points are those with binary preference equal to 1. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the final binary embedding (last column). The different rows provide an idea on the impact of the different possible choices (the type of sampling, parameter σ in Eq. (2), type of classifier). The obtained classifiers can be very different, thus able to characterize clusters with different shapes or structures. This is evident when looking at the clustering results, which are reported in Fig. 4 (bottom right plot); in that figure, we also reported the results obtained by other classical clustering methods, namely the classical hierarchical clustering and K-means, as well as more recent and advanced approaches, namely *affinity propagation* [57] and *spectral clustering* (in the version with the unnormalized graph Laplacian [4]). Those techniques may have difficulties in clustering this dataset, since they assume a particular type of shape/structure for the clusters. On the contrary, our approach, which can exploit a library of models to characterize the different structures present, is able to recover the structure of the clusters. For the proposed approach, in this experiment, we used two categories of one-class classifiers (Gaussian model and linear classifier), training 400 models of each type, starting from 5 points sampled using the compact sampling scheme (with $\sigma = 0.05$).

4. Experimental evaluation

In this section, the proposed technique is evaluated on 12 real datasets.[3] the main characteristics of these datasets are shown in Table 1.

To have a large scope analysis, we have selected a collection of datasets covering different numbers of objects (the smallest dataset contains 32 objects, the largest one 768), different num-

[3] The first 10 are from the *UCI ML Repository*– http://archive.ics.uci.edu/ml/. We thank Pablo Mesejo for providing us the last two datasets (Gastro1 and Gastro2). These datasets have been used in [58]: the suffix refers to type of light used (1: White Light or 2: Narrow Band Imaging).

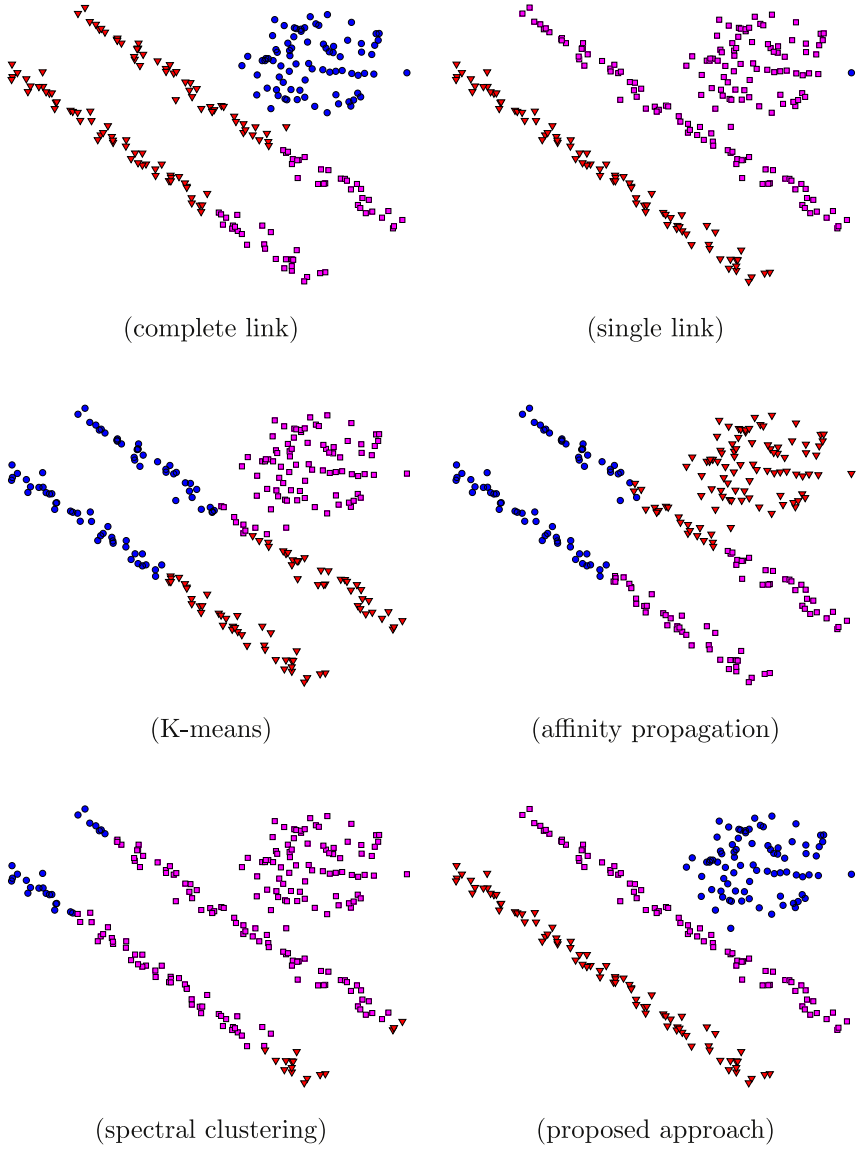

Fig. 4. (Best viewed in color) Toy example: results.

Table 1
Details of the datasets employed for testing.

Name	#objects	#features	#cluster	#obj per cluster
Iris	150	4	3	50,50,50
Ecoli	336	7	8	143,2,77,2,35,20,5,52
Pima	768	8	2	268,500
Glass	214	9	4	70,76,17,51
WBC	683	9	2	444,239
BTissue	106	9	6	21,15,18,16,14,22
Wine	178	13	3	59,71,48
Heart	297	13	2	160,137
Lung	32	54	3	9,13,10
Nose	358	128	5	92,83,32,76,75
Gastro1	76	698	3	21,15,40
Gastro2	76	698	3	21,15,40

bers of features (ranging from 4 to 698), and different cluster sizes (the smallest cluster contains 2 objects, the largest 444). All these datasets are characterized by a limited number of clusters (the maximum is 8); even if it is possible to resort to complex procedures to automatically estimate such number, here we simply provided it in input to all the algorithms. All the datasets were standardized, an operation that is crucial for many clustering approaches. For some datasets, however, this operation lead to a drastic reduction of the clustering accuracies for all techniques (for the proposed approach as well as for the competitors); in such cases, we operated on the original data.

In the experiments herein reported, our method used binary embeddings based on the one-class 1-NN model [21,59]: this scheme is derived from a local density estimation of the data by the nearest neighbor classifier; the method is suitable also for high dimensional spaces, since it avoids explicit density estimation and only uses distances to the first nearest neighbor. More in detail, for a given point, its membership to the one-class model is computed as its distance to the nearest neighbor in the training set, normalized by the distance from this training object to its nearest neighbour (for more details see [21,59]). Despite its simplicity, this rule allows obtaining non-linear decision boundaries (see Fig. 3(e)) and to scales well to moderately high-dimensional spaces.

For a given experiment, we built the embedding space by using 100 models ($M = 100$), i.e., we sampled 100 subsets of points. We used both *compact* and *elongated* strategies; moreover, we also experimented an hybrid scheme, obtained by sampling $M/2$ sets of points with the *compact* sampling scheme and $M/2$ sets with the

Table 2
Clustering results, for different datasets and different versions of the proposed framework, using Purity and Adjusted Rand Index.

Type of sampling: Compact – *Purity*												
Model	Iris	Ecoli	Pima	Glass	WBC	BTissue	Wine	Heart	Lung	Nose	Gastro1	Gastro2
BerMix (Sel)	0.893	0.743	0.668	0.671	0.938	0.667	0.938	0.716	0.485	0.818	0.613	0.560
BerMix (Ens)	0.899	0.758	0.668	0.676	0.938	0.638	0.876	0.841	0.581	0.784	0.613	0.693
CRAFT (Sel)	0.893	0.454	0.657	0.587	0.943	0.514	0.932	0.726	0.485	0.765	0.600	0.547
CRAFT (Ens)	0.899	0.502	0.686	0.577	0.933	0.657	0.881	0.780	0.614	0.700	0.613	0.693
Type of sampling: Compact – *Adjusted Rand Index (ARI)*												
Model	Iris	Ecoli	Pima	Glass	WBC	BTissue	Wine	Heart	Lung	Nose	Gastro1	Gastro2
BerMix (Sel)	0.584	0.404	0.094	0.297	0.768	0.453	0.816	0.047	0.097	0.644	0.134	0.026
BerMix (Ens)	0.654	0.632	0.105	0.303	0.768	0.316	0.690	0.198	0.188	0.359	0.139	0.222
CRAFT (Sel)	0.544	0.090	0.060	0.175	0.784	0.267	0.798	-0.005	0.039	0.548	0.095	0.044
CRAFT (Ens)	0.835	0.716	0.110	0.255	0.748	0.310	0.690	0.015	0.228	0.323	0.118	0.222
Type of sampling: Elongated – *Purity*												
Model	Iris	Ecoli	Pima	Glass	WBC	BTissue	Wine	Heart	Lung	Nose	Gastro1	Gastro2
BerMix (Sel)	0.926	0.851	0.674	0.657	0.922	0.600	0.955	0.750	0.646	0.821	0.560	0.573
BerMix (Ens)	0.899	0.782	0.670	0.620	0.933	0.581	0.915	0.784	0.581	0.745	0.533	0.587
CRAFT (Sel)	0.960	0.457	0.679	0.653	0.925	0.619	0.960	0.740	0.614	0.770	0.627	0.547
CRAFT (Ens)	0.899	0.588	0.683	0.582	0.933	0.619	0.966	0.780	0.614	0.658	0.627	0.693
Type of sampling: Elongated – *Adjusted Rand Index (ARI)*												
Model	Iris	Ecoli	Pima	Glass	WBC	BTissue	Wine	Heart	Lung	Nose	Gastro1	Gastro2
BerMix (Sel)	0.525	0.348	0.100	0.294	0.713	0.396	0.867	0.050	0.224	0.655	0.094	0.045
BerMix (Ens)	0.674	0.684	0.106	0.286	0.748	0.367	0.901	0.329	0.185	0.458	0.061	0.070
CRAFT (Sel)	0.742	0.075	0.108	0.284	0.723	0.328	0.884	-0.005	0.251	0.541	0.132	0.043
CRAFT (Ens)	0.696	0.764	0.130	0.283	0.748	0.308	0.883	0.394	0.261	0.496	0.139	0.220
Type of sampling: "No Choice" – *Purity*												
Model	Iris	Ecoli	Pima	Glass	WBC	BTissue	Wine	Heart	Lung	Nose	Gastro1	Gastro2
BerMix (Sel)	0.893	0.770	0.661	0.685	0.941	0.562	0.842	0.777	0.581	0.779	0.667	0.613
BerMix (Ens)	0.899	0.779	0.666	0.610	0.933	0.581	0.915	0.774	0.581	0.678	0.533	0.587
CRAFT (Sel)	0.893	0.466	0.708	0.620	0.940	0.591	0.842	0.777	0.549	0.776	0.573	0.573
CRAFT (Ens)	0.899	0.466	0.692	0.582	0.931	0.591	0.977	0.777	0.614	0.678	0.600	0.707
Type of sampling: "No Choice" – *Adjusted Rand Index (ARI)*												
Model	Iris	Ecoli	Pima	Glass	WBC	BTissue	Wine	Heart	Lung	Nose	Gastro1	Gastro2
BerMix (Sel)	0.771	0.424	0.084	0.313	0.778	0.319	0.598	-0.002	0.153	0.512	0.234	0.092
BerMix (Ens)	0.654	0.610	0.106	0.294	0.748	0.367	0.748	0.345	0.167	0.375	0.073	0.062
CRAFT (Sel)	0.886	0.073	0.165	0.234	0.773	0.305	0.619	0.027	0.108	0.573	0.082	0.065
CRAFT (Ens)	0.644	0.720	0.118	0.172	0.743	0.316	0.931	0.277	0.261	0.367	0.101	0.263

elongated scheme. This increases the variety of the library of classifiers; we call this scheme the "no-choice" option.

The adopted sampling strategies require choosing two parameters (σ and N_P), which may be different depending on the given dataset. In principle, the choice of these parameters is not very critical, since they simply provide "guidelines" for the sampling of random points used to build the one-class classifiers. In practice, due to the limited number of models, this choice is crucial, and may affect the final result. As a general guideline, we can say that a preliminary evaluation (not presented here) has shown that small values of σ (in the range [0.1-0.9]) should be preferred, independently of the dataset. This is not surprising, since small values lead to sets of nearby points, which characterize a limited region. On the contrary, the impact of N_P is less crucial. In our experiments, we set σ to 0.1 for Iris, Ecoli, and Glass, to 0.3 for BTissue and Heart, to 0.5 for WBC, Wine, and Gastro2, 0.9 for Gastro2 and 0.7 for the remaining datasets. Parameter N_P was set to 10 for Iris, Pima, Glass, Wine, and Nose, and to 15 for the others.

For clustering the embeddings, we used both the Bernoulli mixture and the CRAFT approaches described in Section 3. In particular, the Bernoulli mixture was initialized using a run of the K-means++ algorithm (where we substituted the Euclidean distance with the Hamming distance); the EM algorithm is stopped when the relative change in log-likelihood falls below a threshold. In order to decrease the dependence of the result on the K-means++ initialization (which can be particularly problematic in case of a small number of models), we repeated the process 15 times, keeping the mixture with the highest value of the log-likelihood. Concerning CRAFT, we used the implementation released by the authors,[4] keeping all the parameters at their default values. We repeated the training 15 times, keeping the clustering with the best value of the objective function that underlies the method.

The whole process (sampling, embedding and clustering) is repeated 50 times (i.e. $L = 50$), and the final clustering is obtained by using the two procedures described in Section 3.5 (*Sel* and *Ens*).

For all the datasets, the quality of the clustering results was assessed using the purity index [60] and the *adjusted Rand index* (ARI) [61–63], two classical measures of clustering quality. To compute purity, each cluster is assigned to the class label that is most frequent in that cluster. Purity corresponds to the proportion of examples assigned to the correct label; it lies between 0 (worst) and 1 (best). To compute the ARI, we first build a contingency table between the clustering and the true labeling; then the ARI is derived by measuring the agreement between the two partitions (the Rand index) corrected for the chance of the formation of the clusters. Also in this case, the higher the index value, the better the clustering. All the results are reported in Table 2.

Different observations can be made about the results in Table 2 and about the whole set of experiments. In particular, we focus on three different questions: (i) which is the best clustering method: Bernoulli mixture or CRAFT? (ii) which is the best type of sampling: elongated, compact, or both? (iii) finally, which is the best scheme for aggregating the different clustering results: selection via clustering entropy or clustering ensemble? A summary of findings is reported in Table 3. In particular, in every entry, we report the investigated question, and the result of a paired *t*-

[4] We thank Vikas Garg for providing the code.

Table 3
Analysis of results.

Aspect	Result	p-value	# Exp
1. All results: better Bernoulli mixture or CRAFT?	Equal	6.25e−02	3300
2. Automatic results: better Bernoulli mixture or CRAFT?	Equal	5.49e−01	132
3. All results: better compact or elongated sampling?	Equal	1.48e−01	2200
4. All results: better single or multi sampling?	Better single	9.58e−82	2200
5. Automatic results: better Selection (Sel) or Ensemble (Ens)?	Equal	2.33e−01	132

Table 4
Comparison with alternative approaches using the purity index: Kmeans ("K-means"), Kmeans++ ("K-means++"), agglomerative clustering with Single Link Scheme ("HierCl-SL"), Complete Link ("HierCl-CL") and Ward Link ("HierCl-WL"), Gaussian Mixture Models with diagonal, full and spherical covariance matrix ("GMM (diag)", "GMM (full)" and "GMM (spher)", respectively), Spectral clustering with unnormalized graph Laplacian ("SpectClus"), and with the normalized graph Laplacians in the version of Shi-Malik ("SpectClus (SM)") and Jordan-Weiss ("SpectClus (JW)"), and affinity propagation ("AffProp"). Finally, "BEC" refers to the proposed *binary embedding clustering* approach. The best result in each column is shown in bold.

Purity												
Method	Iris	Ecoli	Pima	Glass	WBC	BTissue	Wine	Heart	Lung	Nose	Gastro1	Gastro2
HierCl-SL	0.664	0.454	0.652	0.366	0.651	0.276	0.396	0.537	0.453	0.263	0.533	0.520
HierCl-CL	0.879	0.758	0.651	0.493	0.906	0.467	0.836	0.537	0.549	0.552	0.560	0.520
HierCl-WL	0.886	0.815	0.675	0.516	0.968	0.552	0.927	0.682	0.581	0.703	0.547	0.520
K-means	0.886	0.797	0.660	0.540	0.960	0.572	0.966	0.831	0.549	0.650	0.547	0.520
GMM (Diag)	0.906	0.424	0.651	0.352	0.650	0.572	0.972	0.709	0.388	0.644	0.520	0.520
GMM (Full)	**0.966**	0.445	0.651	0.531	0.849	0.591	**0.977**	0.709	0.517	0.255	0.520	0.520
GMM (Spher)	0.886	0.424	0.651	0.582	0.946	0.200	0.960	0.720	0.614	0.569	0.520	0.520
SpectClus	0.671	0.445	0.651	0.366	0.651	0.276	0.396	0.537	0.453	0.263	0.520	0.520
SpectClus (SM)	0.725	0.600	0.651	0.366	0.940	0.400	0.396	0.537	0.453	0.398	0.520	0.520
SpectClus (JW)	0.812	0.809	0.678	0.493	**0.968**	0.448	0.396	0.537	0.549	0.599	0.520	0.520
K-means++	0.886	0.830	0.660	0.577	0.962	0.562	0.966	0.831	**0.646**	0.650	0.547	0.520
AffProp	0.899	0.839	0.651	0.573	0.959	0.591	0.910	0.811	0.517	0.784	0.533	0.573
BEC	0.960	**0.851**	**0.708**	**0.685**	0.943	**0.667**	**0.977**	**0.841**	**0.646**	**0.821**	**0.667**	**0.707**
Adjusted Rand Index (ARI)												
Method	Iris	Ecoli	Pima	Glass	WBC	BTissue	Wine	Heart	Lung	Nose	Gastro1	Gastro2
HierCl-SL	0.564	0.040	0.002	0.003	0.003	0.001	−0.007	−0.001	0.016	−0.002	0.005	−0.007
HierCl-CL	0.642	0.763	−0.000	0.131	0.653	0.263	0.577	−0.001	0.169	0.180	0.073	0.047
HierCl-WL	0.731	0.518	0.100	0.180	0.874	0.240	0.790	0.132	0.163	0.473	0.071	0.002
K-means	0.730	0.529	0.074	0.206	0.846	0.309	0.897	0.438	0.169	0.389	0.071	0.019
GMM (Diag)	0.834	0.000	0.000	0.000	0.000	0.000	0.915	0.174	0.000	0.376	0.000	0.000
GMM (Full)	**0.904**	0.007	0.001	0.209	0.473	0.345	0.928	0.174	0.042	0.000	0.000	0.000
GMM (Spher)	0.730	0.000	0.055	0.227	0.794	0.399	0.879	0.192	0.167	0.327	0.000	0.000
SpectClus	0.564	0.039	−0.001	0.003	0.003	−0.005	−0.007	−0.001	0.015	−0.002	−0.021	−0.020
SpectClus (SM)	0.564	0.675	−0.001	0.003	0.771	0.059	−0.007	−0.001	0.015	0.061	0.009	0.041
SpectClus (JW)	0.564	0.631	−0.001	0.159	**0.874**	0.330	−0.007	−0.001	0.114	0.365	−0.016	−0.001
K-means++	0.730	0.495	0.074	0.225	0.852	0.285	0.897	**0.438**	0.219	0.389	0.071	0.019
AffProp	0.802	0.507	0.024	0.227	0.841	0.295	0.741	0.386	0.067	0.595	0.074	0.115
BEC	0.886	**0.764**	**0.165**	**0.313**	0.784	**0.453**	**0.931**	0.394	**0.261**	**0.655**	**0.234**	**0.263**

test (with a significance level of 5%) employed to statistically compare the possible alternatives among all possible clustering results where they have been applied on the same conditions. The number of such experiments is reported in the last column of the table, and clearly depends on the investigated aspect: to give an example, we can compare Bernoulli mixture and CRAFT on 1800 experiments (12 datasets × 50 repetitions × 3 sampling strategies), or the compact and elongated sampling strategies on 1200 experiments (12 datasets × 50 repetitions × 2 clustering approaches). These numbers are then doubled since the comparison is done using both purities and ARI values. The result of such *t*-tests is reported in the second column, whereas the corresponding *p*-value is displayed in the third column.

From Table 3, it can be concluded that the Bernoulli mixture and the CRAFT approach are equivalent if we consider all the runs (question 1) or the results obtained with the automatic procedure (question 2). Notice that in this test we excluded the Ecoli dataset, since CRAFT completely failed here — see Table 2; when including this dataset the Bernoulli mixture is consistently better than CRAFT. Concerning the type of sampling, the two strategies seem to be equivalent (question 3); it also clear that there is no advantage in using them simultaneously (question 4). Finally, the two automatic procedures seem to be equivalent (question 5).

4.1. Comparison with alternative clustering methods

This section reports results obtained with other well-known clustering techniques, including classical approaches such as k-means, mixtures of Gaussians, and hierarchical clustering, as well as more recent approaches, such as *affinity propagation* [57], k-means++[5] [43], and *spectral clustering* [4]. For k-means and agglomerative clustering (single link, complete link, and Ward link), we used the versions implemented in Matlab, whereas for Gaussian mixtures we employed the implementation from the Netlab toolbox.[6] finally, the code for affinity propagation was downloaded from the authors' web site.[7]

In the k-means methods (k-means and k-means++), we repeated the clustering 20 times, by using different initializations (random

[5] https://it.mathworks.com/matlabcentral/fileexchange/28804-k-means++

[6] Available from http://www.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/downloads/.

[7] http://www.psi.toronto.edu

for k-means, as described in [43] for k-means++), and retaining the best result (in terms of objective function). The agglomerative clustering methods, single link ("HierCl-SL"), complete link ("HierCl-CL") and Ward link ("HierCl-WL"), were applied with Euclidean distances. In the Gaussian mixture models, we used three versions: "GMM (diag)", with diagonal covariance matrices, "GMM (full)", with full covariance matrices, and "GMM (spher)", with spherical covariance matrices. In all versions we initialized the EM with 5 iterations of k-means, stopping the procedure at likelihood convergence. We used three versions of spectral clustering, one with the unnormalized graph Laplacian ("SpectClus"), and two using normalized graph Laplacians, in the version of Shi-Malik ("SpectClus (SM)") and Jordan-Weiss ("SpectClus (JW)"). Finally, for affinity propagation ("AffProp"), we employed the version which allows setting the number of clusters.

The results are shown in Table 4, for the purity and ARI criteria. We can conclude that the proposed approach compares very well with alternative techniques, always ranking among the best performers and in many cases yielding the best result. The biggest advantage of the proposed method over the alternatives herein considered is found in the higher dimensional datasets, highlighting known difficulty of classical clustering techniques to operate in high-dimensional spaces (see, for example, the discussions in [5]).

5. Conclusions

In this paper, we proposed a novel clustering scheme based on binary embedding. The proposed approach defines the binary embedding of an object as the output of a collection of one-class classifiers, each learned from a small subset of randomly selected points of the dataset to be clustered. The binary signatures are then clustered using a mixture of Bernoulli distributions or a recently proposed binary biclustering approach called CRAFT. Empirical results confirm the suitability of the proposed scheme in comparison with state-of-the-art alternatives. The main feature of the proposed method is that it is agnostic to the shape of the clusters: this feature can be useful also in other unsupervised scenario, such as biclustering. In this case, however, a proper way to recover memberships of features is required: this aspect is currently under investigation.

Acknowledgments

M. Bicego was partially supported by the University of Verona through the program "Bando di Ateneo per la Ricerca di Base2015".

References

[1] A.K. Jain, R. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.
[2] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[3] A.K. Jain, Data clustering: 50 years beyond k-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[4] U. von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.
[5] F. Nie, Z. Zeng, I.W. Tsang, D. Xu, C. Zhang, Spectral embedded clustering: a framework for in-sample and out-of-sample spectral clustering, IEEE Trans. Neural Netw. 22 (11) (2011) 1796–1808.
[6] G. McLachlan, D. Peel, Finite Mixture Models, John Wiley and Sons, 2000.
[7] M. Filippone, F. Camastra, F. Masullia, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern Recognit. 41 (1) (2008) 176–190.
[8] M. Bicego, M.A.T. Figueiredo, Soft clustering using weighted one class support vector machines, Pattern Recognit. 42 (1) (2009) 27–32.
[9] F. Nie, X. Wang, H. Huang, Clustering and projected clustering with adaptive neighbors, in: Proceedings of the Twentieth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 977–986.
[10] F. Nie, C. Ding, D. Luo, H. Huang, Improved minmax cut graph clustering with nonnegative relaxation, in: Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, 2010, pp. 451–466.
[11] F. Nie, X. Wang, M.I. Jordan, H. Huang, The constrained Laplacian rank algorithm for graph-based clustering, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 1969–1976.
[12] M.S. Charikar, Similarity estimation techniques from rounding algorithms, in: Proceedings of the ACM Symposium on Theory of Computing, 2002, pp. 380–388.
[13] M. Raginsky, S. Lazebnik, Locality-sensitive binary codes from shift-invariant kernels, in: Proceedings of the Advances in Neural Information Processing Systems, 2009, pp. 1509–1517.
[14] B. Kulis, T. Darrell, Learning to hash with binary reconstructive embeddings, in: Proceedings of the Advances in Neural Information Processing Systems, 2009, pp. 1042–1050.
[15] M. Norouzi, D. Fleet, Minimal loss hashing for compact binary codes, in: Proceedings of the International Conference on Machine Learning, 2012, pp. 353–360.
[16] Y. Gong, S. Lazebnik, A. Gordo, F. Perronnin, Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 35 (12) (2013) 2916–2929.
[17] X. Yi, C. Caramanis, E. Price, Binary embedding: fundamental limits and fast algorithm, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 2162–2170.
[18] Y. Gong, S. Kumar, H.A. Rowley, S. Lazebnik, Learning binary codes for high-dimensional data using bilinear projections, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2013, pp. 484–491.
[19] F.X. Yu, S. Kumar, Y. Gong, S.F. Chang, Circulant binary embedding, in: Proceedings of the International Conference on Machine Learning, 2014, pp. 946–954.
[20] M. Moya, D. Hush, Network constraints and multi-objective optimization for one-class classification, Neural Netw. 9 (1996) 463–474.
[21] D.M.J. Tax, One-class classification, Delft University of Technology, Delft, 2001 Ph.D..
[22] O. Mazhelis, One-class classifiers: a review and analysis of suitability in the context of mobile-masquerader detection, South Afr. Comput. J. 36 (2006) 29–48.
[23] S.C. Madeira, A.L. Oliveira, Biclustering algorithms for biological data analysis: a survey, IEEE Trans. Comput. Biol. Bioinform. 1 (2004) 24–44.
[24] M. Law, M. Figueiredo, A.K. Jain, Simultaneous feature selection and clustering using a mixture model, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 1154–1166.
[25] V.K. Garg, C. Rudin, T.S. Jaakkola, CRAFT: Cluster-specific assorted feature selection, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2016, pp. 305–313.
[26] C. Lai, D.M.J. Tax, R.P.W. Duin, E. Pekalska, P. Paclik, A study on combining image representations for image classification and retrieval, Int. J. Pattern Recognit. Artif. Intell. 18 (5) (2004) 867–890.
[27] M. Bicego, V. Murino, M.A.T. Figueiredo, Similarity-based classification of sequences using hidden Markov models, Pattern Recognit. 37 (12) (2004) 2281–2291.
[28] M. Orozco-Alzate, R.P.W. Duin, G. Castellanos-Domínguez, A generalization of dissimilarity representations using feature lines and feature planes, Pattern Recognit. Lett. 30 (3) (2009) 242–254.
[29] E. Pekalska, R.P.W. Duin, The Dissimilarity Representation for pattern Recognition - Foundations and Applications, World Scientific, 2005.
[30] R. Toldo, A. Fusiello, Robust multiple structures estimation with j-linkage, in: Proceedings of the European Conference on Computer Vision, 2008, pp. 537–547.
[31] R. Toldo, A. Fusiello, Image-consistent patches from unstructured points with j-linkage, Image Vis. Comput. 31 (10) (2013) 756–770.
[32] L. Magri, A. Fusiello, T-linkage: a continuous relaxation of j-linkage for multi-model fitting, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2014, pp. 3954–3961.
[33] M. Zuliani, C.S. Kenney, B.S. Manjunath, The multiransac algorithm and its application to detect planar homographies, in: Proceedings of the International Conference on Image Processing, 2005, pp. 153–156.
[34] L. Xu, E. Oja, P. Kultanen, A new curve detection method: randomized hough transform (RHT), Pattern Recognit. Lett. 11 (5) (1990) 331–338.
[35] S. Dasgupta, Experiments with random projection, in: Proceedings of the International Conference on Uncertainty in Artificial Intelligence, 2000, pp. 143–151.
[36] X.Z. Fern, C.E. Brodley, Random projection for high dimensional data clustering: a cluster ensemble approach, in: Proceedings of the International Conference on Machine Learning, 2003, pp. 186–193.
[37] C. Boutsidis, A. Zouzias, P. Drineas, Random projections for *k*-means clustering, in: Advances in Neural Information Processing Systems, 2010, pp. 298–306.
[38] A. Cardoso, A. Wichert, Iterative random projections for high-dimensional data clustering, Pattern Recognit. Lett. 33 (13) (2012) 1749–1755.
[39] S.K. Tasoulis, D.K. Tasoulis, V.P. Plagianakos, Random direction divisive clustering, Pattern Recognit. Lett. 34 (2) (2013) 131–139.
[40] K. Zhao, A.A.A. Wiliem, B.C. Lovell, Efficient clustering on Riemannian manifolds: a kernelised random projection approach, Pattern Recognit. 51 (2016) 333–345.
[41] A.L.N. Fred, A.K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Trans. Pattern Anal. Mach. Intell. 27 (6) (2005).
[42] F. De la Torre, T. Kanade, Discriminative cluster analysis, in: Proceedings of the International Conference on Machine Learning, 2006, pp. 241–248.
[43] D. Arthur, S. Vassilvitskii, K-means++: The advantages of careful seeding, in: Proceedings of the ACM-SIAM Symposium on Discrete algorithms, 2007, pp. 1027–1035.
[44] D.M.J. Tax, R.P.W. Duin, Support vector domain description, Pattern Recognit. Lett. 20 (11–13) (1999) 1191–1199.

[45] M. Bicego, M.A.T. Figueiredo, Soft clustering using weighted one-class support vector machines, Pattern Recognit. 42 (2009) 27–32.
[46] G. Govaert, M. Nadif, Comparison of the mixture and the classification maximum likelihood in cluster analysis with binary data, Comput. Stat. Data Anal. 23 (1) (1996) 65–81.
[47] G. Govaert, M. Nadif, Clustering with block mixture models, Pattern Recognit. 36 (2) (2003) 463–473.
[48] G. Govaert, M. Nadif, An EM algorithm for the block mixture model, IEEE Trans. Pattern Anal. Mach. Intell. 27 (4) (2005) 643–647.
[49] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Stat. Soc. B 39 (1977) 1–38.
[50] C.F.J. Wu, On the convergence properties of the EM algorithm, Ann. Stat. 11 (1) (1983) 95–103.
[51] F. Nie, D. Xu, X. Li, Initialization independent clustering with actively self–training method, IEEE Trans. Syst. Man Cybern. Part B (Cybern.) 42 (1) (2012) 17–27.
[52] B. Kulis, M.I. Jordan, Revisiting k-means: new algorithms via Bayesian nonparametrics, in: Proceedings of the International Conference on Machine Learning, 2012, pp. 1131–1138.
[53] T. Broderick, B. Kulis, M.I. Jordan, Mad-Bayes: Map-based asymptotic derivations from Bayes, in: Proceedings of the International Conference on Machine Learning, 2013, pp. 226–234.
[54] T. Li, S. Ma, M. Ogihara, Entropy-based criterion in categorical clustering, in: Proceedings of the International Conference on Machine Learning, 2004, p. 68.
[55] S. Vega-Pons, J. Ruiz-Shulcloper, A survey of clustering ensemble algorithms, Int. J. Pattern Recognit. Artif. Intell. 25 (03) (2011) 337–372.
[56] B. Hanczar, M. Nadif, Ensemble methods for biclustering tasks, Pattern Recognit. 45 (11) (2012) 3938–3949.
[57] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (2007) 972976.
[58] P. Mesejo, D. Pizarro, A. Abergel, O. Rouquette, S. Beorchia, L. Poincloux, A. Bartoli, Computer-aided classification of gastrointestinal lesions in regular colonoscopy, IEEE Trans. Med. Imaging 35 (9) (2016) 2051–2063.
[59] D. De Ridder, D. Tax, R.P.W. Duin, An experimental comparison of one-class classification methods, in: Proceedings of the Fourth Annual Conference of the Advanced School for Computing and Imaging, 1998, pp. 213–218.
[60] C.D. Manning, P. Raghavan, H. Schutze, Introduction to Information Retrieval, Cambridge University Press, 2008.
[61] L. Hubert, P. Arabie, Comparing partitions, J. Classif. (1985) 193–218.
[62] D. Steinley, Properties of the Hubert–Arabie adjusted rand index, Psychol. Methods 9 (3) (2004) 386.
[63] S. Zhong, J. Ghosh, Generative model-based document clustering: a comparative study, Knowl. Inf. Syst. 8 (3) (2005) 374–384.

Manuele Bicego received his Laurea degree and Ph.D. degree in Computer Science from University of Verona in 1999 and 2003, respectively. From 2004 to 2008 he was at the University of Sassari, in the Computer Vision Lab. Currently he is assistant professor (ricercatore) at the University of Verona, and member of the VIPS (Vision Image Processing & Sound) lab at the Computer Science Department. From June 2009 to February 2011 he was also member of the PLUS (Pattern analysis, Learning and image Understanding Systems) lab at the Istituto Italiano di Tecnologia (IIT - Genova Italy). His research interests include statistical pattern recognition, mainly probabilistic models (GMM, HMM) and kernel machines (e.g. SVM), with application to video analysis, biometrics and, recently, bioinformatics. Manuele Bicego is author of several papers in the above subjects, published in international journals and conferences. He is an associate editor of ELCVIA (Jan 2014 -) and Pattern Recognition (Jul 2016 -). He has served as member of the scientific committee of different international conferences, and he is a reviewer for several international conferences and journals. Manuele Bicego is member of the IEEE Systems, Man, and Cybernetics society and of the IAPR Society Italian Chapter (GIRPR).

Mário A.T. Figueiredo received EE, M.Sc., and Ph.D. degrees in electrical and computer engineering, all from IST, Technical University of Lisbon, Portugal, in 1985, 1990, and 1994, respectively. Since 1994, he has been with the Department of Electrical and Computer Engineering of IST. He is also a researcher and area coordinator at Instituto de Telecomunicações. His interests include statistical pattern recognition, machine learning, image processing, and computer vision. In 1995, he received the Portuguese IBM Scientific Prize. He is/was associate editor of several journals, including the IEEE Transactions on Image Processing and the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI). He was guest co-editor of special issues of IEEE-TPAMI and IEEE Transactions on Signal Processing. He co-chaired the 2001 and 2003 Workshops on Energy Minimization Methods in Computer Vision and Pattern Recognition. He has been a member of program committees of many international conferences, including NIPS, ICML, CVPR, EECV, ICASSP, ICIP, ICPR.