Ensemble clustering with a fuzzy approach

Roberto Avogadri and Giorgio Valentini

DSI, Dipartimento di Scienze dell' Informazione,
Università degli Studi di Milano,
Via Comelico 39, 20135 Milano, Italia.
email: {avogadri,valentini}@dsi.unimi.it

Summary. Ensemble clustering is a novel research field that extends to unsupervised learning the approach originally developed for classification and supervised learning problems. In particular ensemble clustering methods have been developed to improve the robustness and accuracy of clustering algorithms, as well as the ability to capture the structure of complex data. In many clustering applications an example may belong to multiple clusters, and the introduction of fuzzy set theory concepts can improve the level of flexibility needed to model the uncertainty underlying real data in several application domains. In this paper, we propose an unsupervised fuzzy ensemble clustering approach that permit to dispose both of the flexibility of the fuzzy sets and the robustness of the ensemble methods. Our algorithmic scheme can generate different ensemble clustering algorithms that allow to obtain the final consensus clustering both in crisp and fuzzy formats.

1 Introduction

Ensemble clustering methods have been recently proposed to improve the accuracy, stability and robustness of clustering algorithms [1, 2, 3, 4]. They are characterized by many qualities like scalability and parallelism, the ability to capture complex data structure and the robustness regarding the noise [5]. Ensemble methods can combine both different data and different clusterings algorithms.

For instance, ensemble algorithms have been used in data-mining to combine heterogeneous data or to combine data in a distributed environment [6]. Other research lines proposed to combine etherogenous clustering algorithms to generate an overall "consensus" ensemble clustering, in order to exploit the different characteristics of clustering algorithms [7]. By another general approach to ensemble clustering, multiple instances of the data are obtained through "perturbations" of the original data: a clustering algorithm is applied to the multiple perturbed data and the results are combined to achieve the overall ensemble clustering. In this contest several techniques have been proposed, such as noise injection, bagging, random projections [8, 9]. These

methods try to improve both the accuracy and the diversity of each component (base) clustering. In fact several works showed that the diversity among the solutions of the components of the ensemble is one of the crucial factors to develop robust and reliable ensemble clustering algorithms [1] [10].

In many "real world" clustering applications it may occurs that an example may belong to more than one cluster, and in these cases traditional clustering algorithms are not able to capture the real nature of the data. Consider, for instance, general clustering problems in bioinformatics, such as the discovery of functional classes of genes: it is well-known that a single gene may participate to different biological processes, thus it may belong to multiple functional classes of genes. Sometimes it is enough to use hard clustering algorithms and to relax the condition that the final clustering has to be a partition, but more in general different techniques based on probabilistic approaches have been developed (i.e. [3]).

In this contribution we propose an ensemble clustering algorithmic scheme useful to deal with problems where we can capture and manage the possibility for an element to belong to more than one class with different degrees of membership. To achieve this objective we use the fuzzy-set theory to express the uncertainty of the data ownership, and others fuzzy tools to transform the fuzzy clusterings into crisp clusterings. To perturb the data we apply random projections with low distortion [9] , a method well-suited to manage high dimensional data (high number of attributes or "features"), reducing the computational time and improving at the same time the diversity of the data. Combining ensemble clustering techniques and fuzzy set theory, on one hand we can improve the accuracy and the robustness of the consensus ensemble clustering, and on the other hand we can deal with the uncertainty and the fuzziness underlying real data.

In the following sections we introduce the random projections and the fuzzy operators that characterize our proposed unsupervised ensemble methods. Then we describe the fuzzy ensemble clustering algorithmic scheme and the algorithms that can be obtained from it. In sect. 5 we present some results of their application to synthetic and real data sets. The discussion and the conclusions end the paper.

2 Random projections.

Our proposed method applies random projections with low distortion to perturb the data. The objective is to reduce the dimension (number of features) of the data, in order to "preserve" their structure.

Consider a couple of euclidean spaces, the original high d-dimensional, and the target d'-dimensional spaces, with $d > d'$. A random projection θ is a randomized function $\theta : \mathbf{R}^d \longrightarrow \mathbf{R}^{d'}$ such that $\forall p, q \in \mathcal{R}^d$:, $0 < \epsilon < 0.5$, with high probability the following disequalities hold:

$$1 - \epsilon \leq \frac{\|\theta(p) - \theta(q)\|_2}{\|p - q\|_2} \leq 1 + \epsilon$$

An example of random projection is the Plus-Minus-one (PMO) [11] $\theta(p) = R * p$, represented by matrices R, with elements $R_{ij} = 1/\sqrt{d'}(A_{ij})$, where A is a $d' \times d$ matrix and $A_{i,j} \in \{-1, 1\}$ such that $Prob(A_{i,j} = 1) = Prob(A_{i,j} = -1) = 1/2$.

A key problem consists in finding d' such that, for every pair of data $p, q \in \mathbb{R}^d$, the distances between the projections $\theta(p)$ and $\theta(q)$ are approximately preserved with high probability.

A natural measure of the approximation is the distortion $dist_\theta$:

$$dist_\theta(p, q) = \frac{||\theta(p) - \theta(q)||_2}{||p - q||_2} \tag{1}$$

If $dist_\theta(p, q) = 1$, the distances are preserved; if $1 - \epsilon \leq dist_\theta(p, q) \leq 1 + \epsilon$, we say that an ϵ-*distortion* level is introduced.

The main result on random projections is due to the *Johnson-Lindenstrauss (JL) lemma* [12]:
given N vectors $\{x_1, ..., x_N\} \in R^d$, if $d' \geq c\frac{logN}{\epsilon^2}$,, where c is a suitable constant, then it exists a projection $\theta : \mathcal{R}^d \rightarrow \mathcal{R}^{d'}$ that, with high probability, "preserves the distances", for all pairs (x_i, x_j), $i, j \in \{1, \ldots, N\}$;

$$(1 - \epsilon)d(x_i, x_j) \leq d(\theta(x_i), \theta(x_j)) \leq (1 + \epsilon)d(x_i, x_j)$$

If we choose a value of d' according to the JL lemma, we may perturb the data introducing only bounded distortions, approximately preserving the metric structure of the original data (see [13] for more details). Examples of random projections that obey the JL Lemma can be found in [13, 9].
A key note is that the initial space d has to be not "too small", otherwise, using the JL Lemma, the initial space and the reduced space, have to become of similar dimensions, even if the final dimension d' not depends by "d", but only by the number of sample of the data set (n) and by the distortion chosen.

In fact our main target applications are characterized by high dimensionality, such as DNA microarray data [14] , where usually few elements (samples) of high dimensionality (number of features/genes) are available. If $d >> d'$, we can save considerable computational time, working with data set that approximately preserves the metric characteristics of the initial space. The perturbation of the data is obtained randomly choosing for every base learner of the ensemble d' projected features. However different perturbation methods can in principle be used.

3 Fuzzy-set & fuzzy-set methods

3.1 The membership functions

In the classic set theory (called also "crisp set theory"), created by Cantor, the membership values of an object to a set can be 0 (FALSE) or 1 (TRUE). The characteristic function of a crisp set can be defined as follow:

$$I_{crisp\_sets} : \{(elem, crisp\_set)\} \longmapsto \{0, 1\}.$$

The fuzzy set theory [15] is a generalization of the previous theory; in fact an object (elem) can belong only partially to a set (fuzzy_set). It is defined a so called "membership function": $\mu_{fuzzy\_sets} : (elem, fuzzy\_set) \longmapsto [0, 1]$ In general, the domain of a membership function of a fuzzy set U can be every set, but usually it is a discrete set ($U = \{u_1, u_2, ...u_m\}$) or it is a subset of $\mathbf{R}$ ($\mathbf{U}$=[lower_value..higher_value] or $\mathbf{U}$=(lower_value..higher_value), where "lower_value" and "higher_value" can be every real number belong to [0,1], with lower_value < higher_value).
If we consider a fuzzy set A, and a membership function defined on it, we can rewrite the membership function definition as follow:

$$\mu_A : \mathbf{U} \longmapsto [0, 1]$$

so that the membership value $\mu_A(u_i)$ describes the degree of ownership of the element u_i to the set A.

3.2 Fuzzy methods

In several data clustering applications, it is useful to have a method that can capture with a certain approximation the "real" structure of data to obtain the "best" clustering. Through the fuzzy ensemble clustering algorithm we propose, it is possible to manage not only the possibility of overlapping among the clusters, but also the degree of membership of every example of the data set to the different clusters. In some application, however the initial problem does not admit a strictly fuzzy answer, but at the same time it is generally useful to have a valuation method that can use all the possible information available (like the degrees of membership). We use two classical methods to "defuzzify" the results:

1. the "alpha-cut";
2. the "hard-clustering".

The "alpha-cut" function can be defined as follow.
$\forall \alpha \in [0, 1]$ the α-*cut* $[A]_\alpha$, or simply A_α, of A is:

$$[A]_\alpha = \{u \in \mathbf{U} | \mu_A(u) \geq \alpha\}.$$

The expression of a threshold value for the membership function allows to obtain from a fuzzy set a crisp set, called A_α, which contains every element of U whose membership to A is larger than α.
The "hard-clustering" is not properly a fuzzy function: it's a method to obtain a crisp clustering from the original fuzzy clustering.
The role of both the previous functions in the algorithm scheme will be described with more details in Section 4.

3.3 Triangular norms

To generalize the "classical" intersection operator in the fuzzy logic are often used the so called triangular norms (t-norms) [16].
A t-norm T is a function $T : [0,1] \times [0,1] \rightarrow [0,1]$ that satisfies:

- 1) the boundaries conditions: $T(0,0) = T(0,1) = T(1,0) = 0$
- 2) the identity condition: $T(a,1) = a \ \forall$ a $\in (0,1]$
- 3) the commutative property: $T(a,b) = T(b,a) \ \forall$ (a,b) $\in [0,1]$
- 4) the monotonic property: $T(a,b) \leq$ T(c,d) if $a \leq c$ e $b \leq d \ \forall$ (a,b,c,d) $\in$ [0,1]
- 5) the associative property: $T(a,T(b,c)) = T(T(a,b),c) \ \forall$ (a,b,c) $\in [0,1]$

In the literature have been proposed four basic t-norms:

$$T_M(x,y) = min(x,y), (minimum) \tag{2}$$

$$T_P(x,y) = x * y, (algebraic\ product) \tag{3}$$

$$T_L(x,y) = max(x+y-1,0), (Lukasewicz's\ t\text{-}norm) \tag{4}$$

$$T_D(x,y) = \begin{cases} 0 & \text{if } (x,y) \in [0,1)^2, \\ min(x,y) & \text{otherwise.} \end{cases} (drastic\ product) \tag{5}$$

The following order relation exist among the previous t-norms:

$$T_D < T_L < T_P < T_M$$

In our algorithm scheme we used the algebraic product as aggregation operator (see 4).

4 The algorithmic scheme

4.1 General structure

The general structure of the algorithm is similar to the *Randclust* algorithm, proposed in [9]: data are perturbed through random projections to lower dimensional subspaces and multiple clusterings are performed on the projected

data; note that it is likely to obtain different clusterings, since the clustering algorithm is applied to different "views" of the data. Then the clusterings are combined, and a *consensus* ensemble clustering is computed.

The main difference of our proposed method consists in using a fuzzy k-means algorithm as base clustering and in applying a fuzzy approach to the combination and the consensus steps of the ensemble algorithm. In particular we can apply different crisp and fuzzy approaches both to the aggregation and consensus steps, obtaining in this way the following algorithmic scheme:

Fuzzy ensemble clustering algorithmic scheme:

1. *Random projections.* Generation of multiple instances (views) of compressed data through random projections (but different type of data perturbation methods like resampling or noise-injection can also be used).
2. *Generation of multiple fuzzy clusterings.* The fuzzy k-means algorithm is applied to compressed data obtained from the previous step. The output of the algorithm is a membership matrix where each element represents the membership of an example to a particular cluster.
3. *"Crispization" of the base clusterings.* This step is executed if a "crisp" aggregation is performed: the fuzzy clusterings obtained in the previous step can be "defuzzified" through one of the following techniques:
   a) hard-clustering;
   b) α-cut;
4. *Aggregation.* If a fuzzy aggregation is performed, the base clusterings are combined, using a square similarity matrix [8] M^C whose elements are generated through fuzzy t-norms applied to the membership functions of each pair of examples. If a crisp aggregation is performed, the similarity matrix is built using the product of the characteristic function between each pair of examples.
5. *Clustering in the "embedded" similarity space.* The similarity matrix induces a new representation of the data based on the pairwise similarity between pairs of examples: the fuzzy k-means clustering algorithm is applied to the rows (or equivalently to the columns) of the similarity matrix.
6. *Consensus clustering.* The consensus clustering could be represented by the overall consensus membership matrix, resulting in a fuzzy representation of the consensus clustering. Alternatively, we may apply the same crispization techniques used at step 3 to transform the fuzzy consensus clustering to a crisp one.

The two classical "crispization" techniques we used in steps 3 and 6, can be described as follows:

Hard-clustering:

$$\chi^H_{ri} = \begin{cases} 1 \Leftrightarrow \arg\, max_s\, \mathcal{U}_{si} = r \\ 0 \text{ otherwise.} \end{cases} \tag{6}$$

α-cut:

$$\chi_{ri}^{\alpha} = \begin{cases} 1 \Leftrightarrow \mathcal{U}_{ri} \geq \alpha \\ 0 \text{ otherwise.} \end{cases} \tag{7}$$

where χ_{ri} is the characteristic function for the cluster r: that is $\chi_{ri} = 1$ if the i^{th} example belongs to the r^{th} cluster, $\chi_{ri} = 0$ otherwise; $1 \leq s \leq k; 1 \leq i \leq n$, $0 \leq \alpha \leq 1$, and $\mathcal{U}$ is the fuzzy membership matrix obtained by applying the fuzzy k-means algorithm. Note that two different types of membership matrices are considered: at step 3 multiple membership matrices $\mathcal{U}$ are obtained through the application of the fuzzy k-means algorithm to multiple instances of the projected data; at step 6 another membership matrix $\mathcal{U}^C$ (where the superscript C stands for "consensus") is obtained by applying the fuzzy k-means algorithm to the rows of the similarity matrix.

We may observe that considering the possibility of applying crisp or fuzzy methods at steps 3 and 6, we can obtain 9 different algorithms, exploiting different combinations of aggregation and consensus clustering techniques. For instance, combining a fuzzy aggregation with a consensus clustering obtained trough α-cut we obtain from the algorithmic scheme a *fuzzy-alpha* ensemble clustering algorithm, while using a hard-clustering crispization technique for aggregation and a fuzzy consensus we obtain a *max-fuzzy* ensemble clustering.

In the next two sections we discuss the algorithms based on the fuzzy aggregation step (*fuzzy-** clustering ensemble algorithms) and the ones based on the crisp aggregation on the base clusterings (*crisp-** clustering ensembles).

4.2 Fuzzy ensemble clustering with fuzzy aggregation of the base clusterings

The pseudo-code of the algorithm is reported below:

Fuzzy-* ensemble clustering:

`Input`:
- a data set $X = \{x_1, x_2, \ldots, x_n\}$, stored in a $d \times n$ D matrix.
- an integer k (number of clusters)
- an integer c (number of clusterings)
- an integer v (integer used for the normalization of the final clustering)
- the fuzzy k-means clustering algorithm $\mathcal{C}_f$
- a procedure the realizes the randomized map μ
- an integer d' (dimension of the projected subspace)
- a function τ that defines the t-norm

`begin algorithm`

(1) `For each` $i, j \in \{1, \ldots, n\}$ `do` $M_{ij} = 0$

(2) `Repeat for` $t = 1$ `to` c

(3) $R_t =$ `Generate_projection_matrix` (d', μ)

(4) $D_t = R_t \cdot D$

(5) $\mathcal{U}^{(t)} = \mathcal{C}_f(D_t, k, m)$

(6) `For each` $i, j \in \{1, \ldots, n\}$

$M_{ij}^{(t)} = \sum_{s=1}^{k} \tau(\mathcal{U}_{si}^{(t)}, \mathcal{U}_{sj}^{(t)})$
`end repeat`
(7) $M^C = \frac{\sum_{t=1}^{c} M^{(t)}}{v}$
(8) $< A_1, A_2, \ldots, A_k >= \mathcal{C}_f(M^C, k, m)$
`end algorithm`.
`Output`:
- the final clustering $C =< A_1, A_2, \ldots, A_k >$
- the cumulative similarity matrix M^C.

Note that the dimension d' of the projected subspace is an input parameter of the algorithm, but it may be computed according to the *JL* lemma (Sect. 2), to approximately preserve the distances between the examples. Inside the mean loop (steps 2-6) the procedure `Generate_projection_matrix` produces a $d' \times d$ R_t matrix according to a given randomized map μ [9], that it is used to randomly project the original data matrix D into a $d' \times n$ D_t projected data matrix (step 4). In step (5) the fuzzy k-means algorithm $\mathcal{C}_f$ with a given fuzziness m is applied to D_t and a k-clustering represented by its $\mathcal{U}^{(t)}$ membership matrix is achieved. Hence the corresponding similarity matrix $M^{(t)}$ is computed, using a given *t-norm* (step 6). Note that $\mathcal{U}$ is a fuzzy membership matrix (where the rows are clusters and the columns examples). A similar approach has been also proposed in [17].

In (7) the "cumulative" similarity matrix M^C is obtained by averaging across the similarity matrices computed in the main loop. Note the normalization factor $\frac{1}{v}$: it's easy to demonstrate that, for the choice of the algebraic product as t-norm, a suitable choice of v can be the number of clusterings c. Finally, the *consensus* clustering is obtained by applying the fuzzy k-means algorithm to the rows of the similarity matrix M^C (step 8).

4.3 Fuzzy ensemble clustering with crisp aggregation of the base clusterings

In agreement with the algorithm scheme, there are two different methods to "defuzzify" the clusterings:

- hard-clustering;
- α-cut.

Below we provide the pseudo-code for the fuzzy ensemble clustering with "crispization" through hard-clustering.
Max-* ensemble clustering:
`Input`:
- a data set $X = \{x_1, x_2, \ldots, x_n\}$, stored in a $d \times n$ D matrix.
- an integer k (number of clusters)
- an integer c (number of clusterings)
- an integer v (integer used for the normalization of the final clustering)

- the fuzzy k-means clustering algorithm $\mathcal{C}_f$
- a procedure the realizes the randomized map μ
- an integer d' (dimension of the projected subspace)
- a "crispization" algorithm "Crisp"

`begin algorithm`
(1) `For each` $i, j \in \{1, \ldots, n\}$ `do` $M_{ij} = 0$
(2) `Repeat for` $t = 1$ `to` c
(3) $R_t =$ `Generate_projection_matrix` (d', μ)
(4) $D_t = R_t \cdot D$
(5) $\mathcal{U}^{(t)} = \mathcal{C}_f(D_t, k, m)$
(5bis) $\chi^{(t)} = \text{Crisp}(\mathcal{U}^{(t)})$
(6) `For each` $i, j \in \{1, \ldots, n\}$
$M_{ij}^{(t)} = \sum_{s=1}^{k} \chi_{si}^{(t)} * \chi_{sj}^{(t)}$
`end repeat`
(7)$M^c = \frac{\sum_{t=1}^{c} M^{(t)}}{c}$
(8) $< A_1, A_2, \ldots, A_k >= \mathcal{C}_f(M^c, k, m)$
`end algorithm`.
`Output`:
- the final clustering $C =< A_1, A_2, \ldots, A_k >$
- the cumulative similarity matrix M^C.

Different observations can be made about the proposed algorithm:

1. With respect to the previously proposed *fuzzy-** ensemble clustering, it has been introduced a new step (step 5bis) after the creation of the membership matrix of the single clusterings, to obtain the transformation of fuzzy data in crisp ones. After this new step a characteristic matrix $\chi^{(t)}$ is created for every clustering.
2. After this step, the data can be managed like "natural" crisp data. In fact, in the (6) step, the final similarity matrix is obtained through the methods showed in [8, 9].
3. As a consequence of the "hard-clustering crispization" (step 5bis) the consensus clustering is a partition, that is each example may belong to one and only one cluster:

$$\forall i, j \ \chi_{ij}^{t} = \begin{cases} 1 \iff argmax_s \ \mathcal{U}_{sj} = i, \\ 0 \text{ otherwise.} \end{cases}$$

Hence, the normalization of the values of the similarity matrix can be performed using the factor $\frac{1}{c}$ (step (7)).

4.4 Fuzzy ensemble clustering with crisp aggregation and α-cut defuzzification

Another algorithm that can be derived from the algorithmic scheme is based on the crisp aggregation of base clustering using an α-cut defuzzification method (eq. 7). In this case the results strongly depend on the choice of the value of α. Indeed for large values of α in several base clusterings we may have many unassigned examples, while, on the contrary, for small values of α it is likely some examples may belong to multiple clusters.

The pseudo-code of the ensemble algorithm with crispization through α-cut is reported below.

Alpha-* ensemble clustering:

`Input`:
- a data set $X = \{x_1, x_2, \ldots, x_n\}$, stored in a $d \times n$ D matrix.
- an integer k (number of clusters)
- an integer c (number of clusterings)
- a real value α (α-cut value)
- the fuzzy k-means clustering algorithm $\mathcal{C}_f$
- a procedure the realizes the randomized map μ
- an integer d' (dimension of the projected subspace)
- a "crispization" algorithm "Crisp"

`begin algorithm`
(1) `For each` $i, j \in \{1, \ldots, n\}$ `do` $M_{ij} = 0$
(2) `Repeat for` $t = 1$ `to` c
(3) $R_t =$ `Generate_projection_matrix` (d', μ)
(4) $D_t = R_t \cdot D$
(5) $\mathcal{U}^{(t)} = \mathcal{C}_f(D_t, k, m)$
(5bis)$\chi^{(t)} = Crisp_\alpha(\mathcal{U}^{(t)})$;
(6) `For each` $i, j \in \{1, \ldots, n\}$
$M_{ij}^{(t)} = \sum_{s=1}^{k} \chi_{si}^{(t)} * \chi_{sj}^{(t)}$
`end repeat`
(7)$M^c = \frac{\sum_{t=1}^{c} M^{(t)}}{k*c}$
(8) $< A_1, A_2, \ldots, A_k >= \mathcal{C}_f(M^c, k, m)$
`end algorithm`.

`Output`:
- the final clustering $C =< A_1, A_2, \ldots, A_k >$
- the cumulative similarity matrix M^C.

Comparing this algorithm with the *Max-** ensemble clustering (Sect. 4.3) we may note that the main changes are in the steps (5bis) and (7). Indeed, in the step (5bis) the Crisp algorithm has a new parameter: α, the *alpha*-cut threshold value. In particular in the $\chi^{(t)} = Crisp_\alpha(\mathcal{U}^{(t)})$ operation, the assignment of an example to a specific cluster depends on the value of α (a parameter that is given as input to the algorithm):

$$\chi_{ij}^{t} = \begin{cases} 1 \Longleftrightarrow \mathcal{U}_{ij} \geq \alpha, \\ 0 \text{ otherwise.} \end{cases}$$

Note that the normalization method in the step (7) comes from the following observations:

1. In the algorithm proposed it is used only one clustering function that works with a fixed number (k) of clusters for each clustering;
2. For a fixed α:
$$\chi_{si} = 1 \Longleftrightarrow \mu_{si} > \alpha$$
$$\chi_{si} = 0 \Longleftrightarrow \mu_{si} \leq \alpha$$
with $1 \leq s \leq k$;
hence
$$0 \leq \sum_{s=1}^{k} \chi_{si} \leq k$$

Considering that each k-clustering is repeated c times, we can observe that $k * c$ is the total number of clusters across the multiple clusterings. We may use base clusterings with different number of clusters for each execution; in this case the normalization factor v becomes:

$$v = \sum_{t=1}^{c} k_t \tag{8}$$

where k_t is the number of clusters of the t^{th} clustering.

5 Experimental results

In this section we test our proposed fuzzy ensemble algorithms with both synthetic and real data. For all the experiments we used high-dimensional data to test the effectiveness of random projections with this kind of complex data.

5.1 Experiments with synthetic data

Experimental environment

To test the performance of the proposed algorithms, we used a synthetic data generator [9]. Every synthetic data set is composed by 3 clusters with 20 samples each. Every example is 5000-dimensional. Each cluster is distributed according to a spherical gaussian probability distribution with a standard deviation of 3. The first cluster in centered in the null vector 0. The other two clusters are centered in $0.5\mathbf{e}$ and $-0.5\mathbf{e}$, where $\mathbf{e}$ is a vector with all the components equal to 1. We tested 4 of the 9 algorithms developed, two with the hard-clustering method applied to the consensus clustering and two using the α-cut approach:

- *max-max*: hard-clustering applied to both the base clustering and consensus clustering;
- *fuzzy-max*: the aggregation step is fuzzy, the consensus step crisp through hard-clustering;
- *max-alpha*: crisp aggregation by hard-clustering, and crisp consensus through α-cut;
- *fuzzy-alpha*: fuzzy aggregation and crisp consensus through α-cut.

We repeated 20 times the previous four clustering ensemble algorithms using data sets randomly projected to a 410-dimensional feature space (corresponding to a distortion $\epsilon = 0.2$, see Sect. 2). As a baseline clustering algorithm we used the classical fuzzy-k-means, executed on the original data set (without data compression). Since clustering does not univocally associate a label to the examples, but only provides a set of clusters, we evaluated the error by choosing for each clustering the permutation of the classes that best matches the "a priori" known "true" classes. More precisely, considering the following clustering function:

$$f(x) : \mathcal{R}^d \rightarrow \mathcal{P}(\mathcal{Y}), \text{ with } \mathcal{Y} = \{1, \ldots, k\} \tag{9}$$

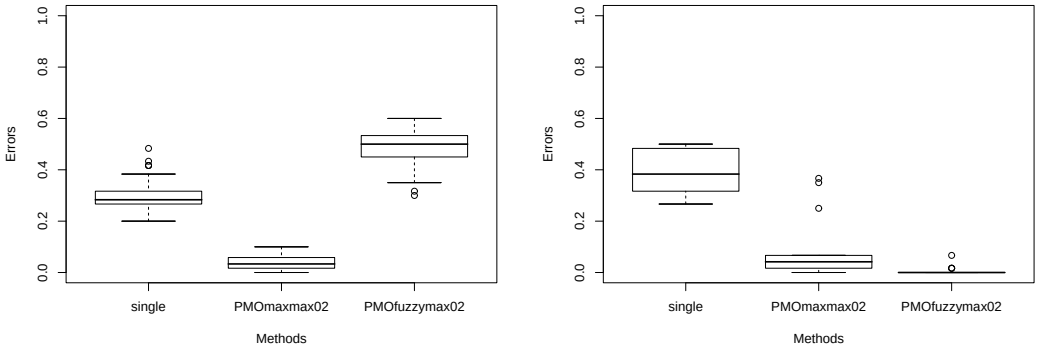
where x is the sample to classify, d its dimension, k the number of the classes, and $\mathcal{P}(\mathcal{Y})$ is the powerset of $\mathcal{Y}$, the error function we applied is the following:

$$\mathcal{L}_{0/1}(Y, t) = \begin{cases} 0 \text{ if } (|Y| = 1 \wedge t \in Y) \vee Y = \{\lambda\} \\ 1 \text{ otherwise.} \end{cases} \tag{10}$$

with t the "real" label of the sample x, $Y \in \mathcal{P}(\mathcal{Y})$ and $\{\lambda\}$ is the empty set. Other loss functions or measures of the performance of clustering algorithms may be applied, but we chose this modification of the 0/1 loss function to take into account the multi-label output of fuzzy k-means algorithms, and considering that our target examples belong only to a single cluster.

Results

The error boxplots of Fig. 1 show that our proposed fuzzy ensemble clustering algorithms perform consistently better than single fuzzy k-means algorithms, with the exception of the fuzzy-max algorithm, when a relatively high level of fuzziness is chosen. More precisely, in Fig.1, we can observe how different degrees of fuzziness of the "component" k-means clusterings can change the performance of the ensemble, if the "fuzzy" information are preserved from the "crispization" operation. In fact, if the single k-means and the ensemble max-max algorithm (which use the hard-clustering operation in both the component and consensus level) performances are similar in the both the graphics (1 (a) and (b)), the result of the fuzzy-max ensemble algorithm (where the "defuzzification" operation are performed only on the final result) change drastically. The good performance of the max-max algorithm with both the



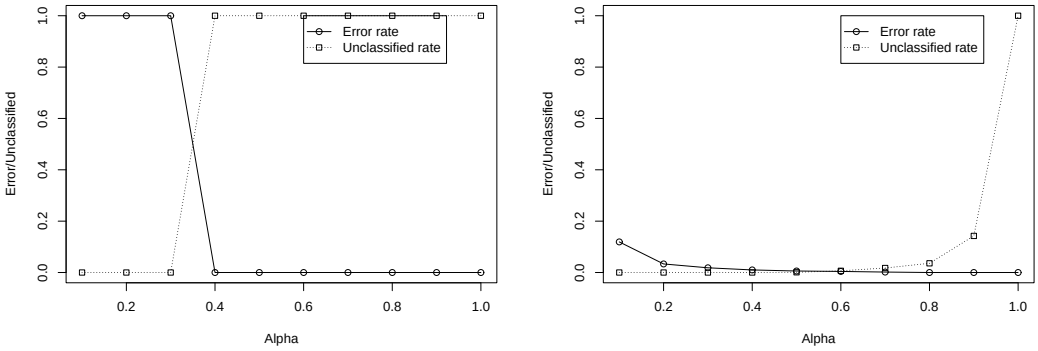
(a) (b)

Fig. 1. Boxplot related the max-max and the fuzzy-max algorithms (PMO data reduction and $\epsilon = 0.2$) compared with the single k-means "crispized" through hard-clustering respectively with (a) fuzziness=2.0 (b) fuzziness=1.1

degrees of fuzziness could depend on the high level of "crispness" of the data: indeed each example is assumed to belong exactly to one cluster. A confirmation of this hypothesis is given by the improvement of performance of the fuzzy-max algorithm by lowering the level of fuzziness (2.0 to 1.1) of the base clusterings. A different consideration can be proposed regarding the fuzzy-max algorithm, in which the capacity to express "pure" fuzzy results on the base learners level, can improve its degree of flexibility (possibility to adapt the algorithm to the nature of the clusterings).

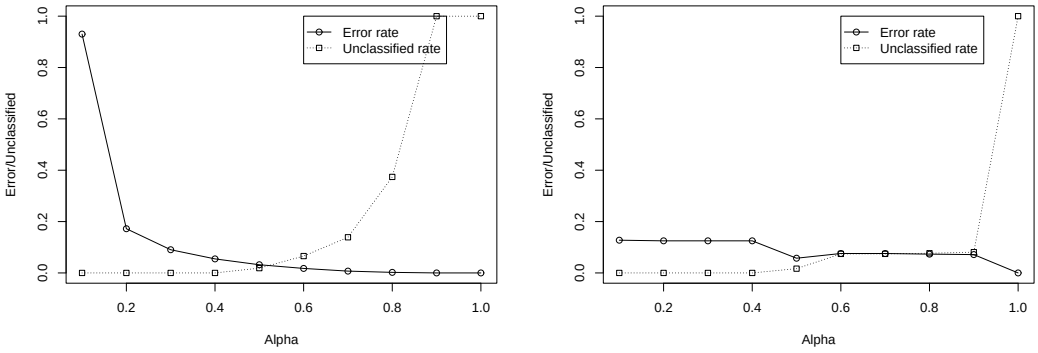
The analysis of the fuzzy-alpha and max-alpha algorithms (Fig. 2, 3) shows how the reduction of the fuzziness reduces the number of unclassified samples and the number of errors, especially for $\alpha \leq 0.5$); for higher level of α the error rate goes to 0, but the number of unclassified samples arises quickly. Note that fuzzy-alpha ensembles achieve an error rate very close to 0 with a small amount of unclassified examples for a large range of α values (Fig. 2 b). Fig 3 shows how the max-alpha algorithm obtains inversely related error and unclassified rates while varying α, with an "optimal" value close to 0.5.

Table 1 summarizes the results, showing that our proposed *fuzzy-max* and *fuzzy-alpha* ensemble methods outperform the other compared ensemble algorithms with respect to these high dimensional synthetic data.



(a) (b)

Fig. 2. *Fuzzy-Alpha* ensemble clustering error and unclassified rate with $fuzziness = 2$ (a) and $fuzziness = 1.1$ (b) with respect to α.



(a) (b)

Fig. 3. *Max-Alpha* ensemble clustering error and unclassified rate with $fuzziness = 2$ (a) and $fuzziness = 1.1$ (b) with respect to α.

5.2 Experiments with real data

Experimental environment

We performed experiments with high dimensional DNA microarray data. In this context each example corresponds to a patient and the features associated with the patients are gene expression levels measured through DNA microarray [14]. These high-throughput bio-technologies allow the parallel measurements of the mRNA levels of thousands of genes (and now of en-

Table 1. Compared results between fuzzy ensemble clustering methods and other ensemble and "single" clustering algorithms. The last column represents the rate of the unclassified examples.

Algorithms	**Mean error**	**Std. Dev.**	**% Uncl.**
Fuzzy-Max	0.0058	0.0155	0
Fuzzy-Alpha	0.0058	0.0155	0.0008
Max-Max	0.0758	0.1104	0
Max-Alpha	0.0573	0.0739	0.0166
Rand-Clust	0.0539	0.0354	0
Fuzzy "single"	0.3916	0.0886	0
Hierarchical "single"	0.0817	0.0298	0

tire genomes) of the cells or tissues of a given patient, thus providing a sort of snapshot of the functional status of a given cell or tissues in a certain condition. In this way we can obtain the molecular portrait of a given phenotype in a given condition and at a given time. Among the different application of this technology, here we consider the analysis of gene expression data of patients to reconstruct known phenotypes using bio-molecular data (DNA microarray measurements). These data are characterized by a high dimension (high number of analyzed genes) and relatively low cardinality (number of patients), thus resulting in a challenging unsupervised problem.

In our experiments we experimented with the *Leukemia* data set, that contains gene expression levels of 7129 genes in Affymetrix's scaled average difference units relative to 47 patients with Acute Lymphoblastic Leukemia (ALL) and 25 cases of Acute Myeloid Leukemia (AML) [18]. We analyzed also the *Melanoma* data set (described in [19]) that it is composed by 31 melanoma samples and 7 control samples with 6971 genes. For both these data sets we applied the same pre-processing procedures described respectively in [18] and [19].

We tested the performance of the fuzzy ensemble algorithms *fuzzy-max*, *fuzzy-alpha*, *max-max* and *max-alpha* using the previously described data sets. We compared the results with *Randclust*, the corresponding "crisp" version of our proposed ensemble methods [9] and with *Bagclust1*, based on an unsupervised version of bagging [8], and with the "single" fuzzy k-means clustering algorithm. The *Randclust* ensemble method is similar to the algorithm presented in this paper, but it uses the hierarchical clustering algorithm to produce the base clusterings, and a crisp approach to combine the resulting clusters. *Bagclust1* generates multiple instances of perturbed data through bootstrap techniques, and then it applies to each instance the base clustering algorithm (we used in our experiments k-means); the final clustering is obtained by majority voting.

Each ensemble is composed by 50 base clusterings and each ensemble method has been repeated 30 times. Regarding to ensemble methods based on random projections, we chose projections with bounded 1 ± 0.2 distor-

tion, according to the *JL* lemma, while for *Bagclust1* we randomly drew with replacement a number of examples equal to the number of the available data.

Results

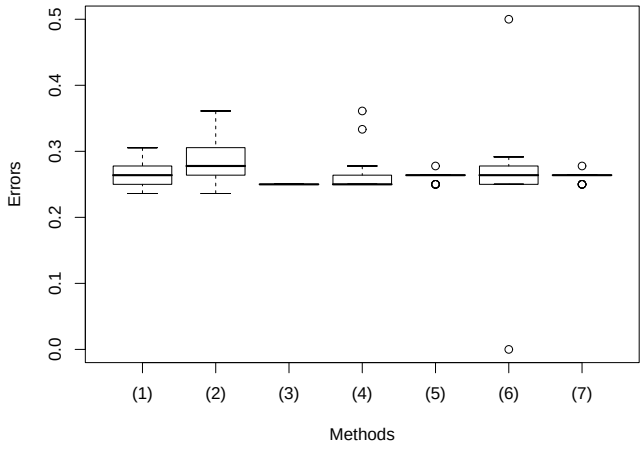
The boxplots in Fig. 4 represent the distribution of the error across multiple repetitions of the fuzzy ensemble algorithms compared with the "single" fuzzy k-means, the *Randclust* and the *Bagclust1* ensemble methods for the two data sets used in the experiments.

With the *Leukemia* data set the results obtained with the different methods are quite comparable (Fig. 4 (a)), while with the *Melanoma* DNA microarray data (Fig. 4 (b)), the proposed fuzzy ensemble methods largely outperform all the other compared methods. The results show that ensembles based on random projection to lower dimensional spaces, using projections matrices that obey the JL lemma are well-suited to high dimensional data. Nevertheless note that in the experiments with the *Leukemia* data set both our proposed fuzzy ensemble clustering method and *Randclust* applied random projections to perturb the data, but our proposed fuzzy approach significantly outperforms the crisp *Randclust* ensemble method.

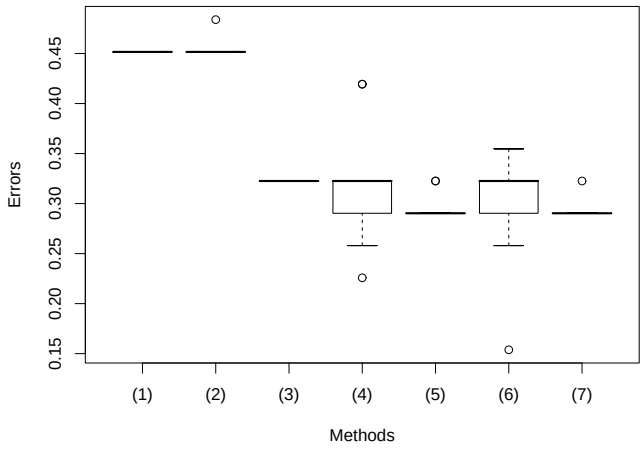
In order to understand in which way the base fuzzy clustering may affect the overall results with respect to the base hierarchical clustering algorithms used in *Randclust*, we performed an analysis of the relationships between accuracy and similarity of each pair of base clusterings used in each ensemble, using measures based on the normalized mutual information (NMI), according to the approach originally proposed in [20] and [1]. By this approach the accuracy of each pair of base clusterings of the ensemble is measured by averaging the NMI of each base clustering with respect to the a priori known "true" clustering, while their diversity by measuring the NMI directly between the two base clusterings. The results are plotted in Fig. 5. Interestingly enough, the base clusterings of our proposed fuzzy ensemble approach are both more accurate (in the figure their NMI ranges approximately between 0.25 and 0.30, while in *Randclust* the accuracy is below 0.20), and more diverse (indeed their NMI in the y axis are between 0.55 and 0.80, while in *Randclust* are above 0.80: recall that a low value of NMI between base clusterings reveals a high diversity and viceversa). It is well-known in the literature that a highly desirable property of ensembles consists in a high accuracy and diversity of base learners: there is a trade-off between accuracy and diversity, and the performances of ensembles in part depend on the relationships between these quantities [4]. Our results show that our proposed ensemble clustering approach improve both the accuracy and the diversity of base learners.

6 Conclusions

In this paper we proposed an algorithmic scheme that combines a fuzzy approach with random projections to obtain clustering ensembles well suited



(a)



(b)

Fig. 4. Boxplot of the results of gene expression data analysis: (a) *Leukemia* data set and (b) *Melanoma* data set. (1)..(7) in abscissa refer to to the results obtained respectively with Randclust (1), Bagclust1 (2), Single fuzzy k-means (3), max-max (4), fuzzy-max (5), max-alpha (6) and fuzzy-alpha (7) fuzzy ensemble algorithms.

to the analysis of complex high-dimensional data. The proposed approach on one hand exploits the accuracy and the effectiveness of the ensemble clustering

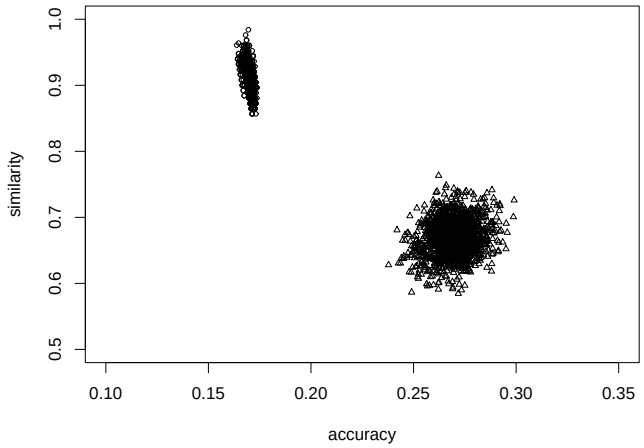


Fig. 5. *Melanoma* data set: analysis of the relationships between accuracy and similarity between the base learners of the fuzzy max* ensemble clustering (triangles) and Randclust ensemble clustering (circles).

techniques based on random projections, and on the other hand the expressive capacity of the fuzzy sets, to obtain clustering algorithms both reliable and able to express the uncertainty of the data.

From the algorithmic scheme several ensemble algorithms can be derived, by combining different fuzzy and defuzzification methods in the aggregation and consensus steps of the general algorithmic scheme. Our preliminary results with both synthetic and DNA microarray data are quite encouraging, showing that the fuzzy approach achieves a good compromise between the accuracy and diversity between the base learners. Moreover these results have been also confirmed by other recent experiments with DNA microarray data [21].

Several open problems need to be considered for future research work. For instance, we may consider the choice of the t-norm to be used in the fuzzy aggregation of multiple clusters. In our experiments we applied the algebraic product, but we need to experiment with other t-norm and we need to analyze their properties to understand what could be the better choice with respect to the characteristics of the data. Moreover we experimented with the PMO random projections, but we need also to experiment with other random projections, such as normal or Achlioptas random projections [11, 13], and we need also to get more theoretical insights into the reasons why random projections work on high dimensional spaces. Another interesting development of this work consists in studying if it possible to embed recently proposed

stability-based methods based on random projections [22, 23] into ensemble clustering methods to steer the construction of the consensus clustering.

In the experiments we used "crisp" data, showing that the proposed method can be successfully applied to analyze this kind of data. We are planning new experiments with examples that may belong to multiple clusters (e.g. unsupervised analysis of functional classes of genes) to show more clearly the effectiveness of the proposed approach. Moreover we plan experiments to analyze the structure of unlabeled data when the boundaries of the clusters are highly uncertain, with very partial memberships of the examples to the clusters.

References

1. Fern, X., Brodley, C.: Random projections for high dimesnional data clustering: A cluster ensemble approach. In Fawcett, T., Mishra, N., eds.: Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), Washington D.C., USA, AAAI Press (2003)
2. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus Clustering: A Resampling-based Method for Class Discovery and Visualization of Gene Expression Microarray Data. Machine Learning **52** (2003) 91–118
3. Topchy, A., Jain, A., Puch, W.: Clustering Ensembles: Models of Consensus and Weak Partitions. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(12) (2005) 1866–1881
4. Kuncheva, L., Vetrov, D.: Evaluation of stability of k-means cluster ensembles with respect to random initialization. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(11) (2006) 1798–1808
5. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience, New York (2004)
6. Strehl, A., Ghosh, J.: Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions. Journal of Machine Learning Research **3** (2002) 583–618
7. Hu, X., Yoo, I.: Cluster ensemble and its applications in gene expression analysis. In: Proc. 2nd Asia-Pacific Bioinformatics Conference, Dunedin, New-Zealand (2004) 297–302
8. Dudoit, S., Fridlyand, J.: Bagging to improve the accuracy of a clustering procedure. Bioinformatics **19**(9) (2003) 1090–1099
9. Bertoni, A., Valentini, G.: Ensembles based on random projections to improve the accuracy of clustering algorithms. In: Neural Nets, WIRN 2005. Volume 3931 of Lecture Notes in Computer Science., Springer (2006) 31–37
10. Hadjitodorov, S., Kuncheva, L., Todorova, L.: Moderate Diversity for Better Cluster Ensembles. Information Fusion **7**(3) (2006) 264–275
11. Achlioptas, D.: Database-friendly random projections: Johnson-lindenstrauss with binary coins. Journal of Comp. & Sys. Sci. **66**(4) (2003) 671–687
12. Johnson, W., Lindenstrauss, J.: Extensions of Lipshitz mapping into Hilbert space. In: Conference in modern analysis and probability. Volume 26 of Contemporary Mathematics., Amer. Math. Soc. (1984) 189–206

13. Bertoni, A., Valentini, G.: Randomized maps for assessing the reliability of patients clusters in DNA microarray data analyses. Artificial Intelligence in Medicine **37**(2) (2006) 85–109
14. Eisen, M., Spellman, P., Brown, P., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. PNAS **95**(25) (1998) 14863–14868
15. Zadeh, L.: Fuzzy sets. Information and Control **8** (1965) 338–353
16. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer Academic (2000)
17. Yang, L., Lv, H., Wang, W.: Soft cluster ensemble based on fuzzy similarity measure. (2006)
18. Golub, T., et al.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science **286** (1999) 531–537
19. Bittner, M., Meltzer, P., Chen, Y., Jiang, Y., Seftor, E., Hendrix, M., Radmacher, M., Simon, R., Yakhini, Z., Ben-Dor, A., Sampas, N., Dougherty, E., Wang, E., Marincola, F., Gooden, C., Lueders, J., Glatfelter, A., Pollock, P., Carpten, J., Gillanders, E., Leja, D., Dietrich, K., Beaudry, C., Berens, M., Alberts, D., Sondak, V.: Molecular classification of malignant melanoma by gene expression profiling. Nature **406** (2000) 536–540
20. Dietterich, T.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. Machine Learning **40**(2) (2000) 139–158
21. Avogadri, R., Valentini, G.: Fuzzy ensemble clustering for DNA microarray data analysis. In: CIBB 2007, The Fourth International Conference on Bioinformatics and Biostatistics. Volume 4578 of Lecture Notes in Computer Science., Springer (2007)
22. Valentini, G.: Clusterv: a tool for assessing the reliability of clusters discovered in DNA microarray data. Bioinformatics **22**(3) (2006) 369–370
23. Bertoni, A., Valentini, G.: Model order selection for bio-molecular data clustering. BMC Bioinformatics **8**(Suppl.3) (2007)