First publ. in: International Journal of Approximate Reasoning 45 (2007), pp. 439-454

Fuzzy clustering in parallel universes

Bernd Wiswedel, Michael R. Berthold *

ALTANA-Chair for Bioinformatics and Information Mining, Department of Computer and Information Science, University of Konstanz, 78457 Konstanz, Germany

Abstract

We present an extension of the fuzzy *c*-Means algorithm, which operates simultaneously on different feature spaces—so-called parallel universes—and also incorporates noise detection. The method assigns membership values of patterns to different universes, which are then adopted throughout the training. This leads to better clustering results since patterns not contributing to clustering in a universe are (completely or partially) ignored. The method also uses an auxiliary universe to capture patterns that do not contribute to any of the clusters in the real universes and therefore are likely to represent noise. The outcome of the algorithm is clusters distributed over different parallel universes, each modeling a particular, potentially overlapping subset of the data and a set of patterns detected as noise. One potential target application of the proposed method is biological data analysis where different descriptors for molecules are available but none of them by itself shows global satisfactory prediction results.

Keywords: Fuzzy clustering; Objective function; Noise handling; Multiple descriptor spaces; Parallel universes

1. Introduction

In recent years, researchers have worked extensively in the field of cluster analysis, which has resulted in a wide range of (fuzzy) clustering algorithms [9,10]. Most of the methods assume the data to be given in a single (mostly numeric) feature space. In some

* Corresponding author.
E-mail addresses: wiswedel@inf.uni-konstanz.de (B. Wiswedel), berthold@inf.uni-konstanz.de (M.R. Berthold).

applications, however, it is common to have multiple representations of the data available. Such applications include biological data analysis, in which, e.g. molecular similarity can be defined in various ways. Fingerprints are the most commonly used similarity measure. A fingerprint in a molecular sense is usually a binary vector, whereby each bit indicates the presence or absence of a molecular feature. The similarity of two compounds can be expressed based on their bit vectors using the Tanimoto coefficient, for example. Other descriptors encode numerical features derived from 3D maps, incorporating the molecular size and shape, hydrophilic and hydrophobic regions quantification, surface charge distribution, etc. [6]. Further similarities involve the comparison of chemical graphs, interatomic distances, and molecular field descriptors. However, it has been shown that often a single descriptor fails to show satisfactory prediction results [16].

Other application domains include web mining where a document can be described based on its content and on anchor texts of hyperlinks pointing to it [4]. 3D objects as used in CAD-catalogues, virtual reality applications, medicine and many other domains can be described, for instance, by various so-called *feature vectors*, i.e. vector of scalars whose cardinalities can easily reach a couple of hundreds. Feature vectors can rely on different statistics of the 3D object, projection methods, volumetric representations obtained by discretizing the object's surface, 2D images, or topological matchings. Bustos et al. [5] provide a survey of feature-based similarity measures of 3D objects.

In the following we denote these multiple representations, i.e. different descriptor spaces, as *Parallel Universes* [14], each of which have representations of all objects of the data set. The challenge that we are facing here is to take advantage of the information encoded in the different universes to find clusters that reside in one or more universes each modeling one particular subset of the data. In this paper, we develop an extended fuzzy c-Means (FCM) algorithm [1] with noise detection that is applicable to parallel universes, by assigning membership values from objects to universes. The optimization of the objective function is similar to the original FCM but also includes the learning of the membership values to compute the impact of objects to universes.

In the next section, we discuss in more detail the concept of parallel universes; Section 3 presents related work. We formulate our new objective function in Section 4, introduce the clustering algorithm in Section 5 and illustrate its usefulness with some numeric examples in Section 6.

2. Parallel universes

We consider parallel universes to be a set of feature spaces for a given set of objects. Each object is assigned a representation in each single universe. Typically, parallel universes encode different properties of the data and thus lead to different measures of similarity. (For instance, similarity of molecular compounds can be based on surface charge distribution or a specific fingerprint representation.) Note, due to these individual measurements they can also show different structural information and therefore exhibit distinctive clustering. This property differs from the problem setting in the so-called *Multi-View Clustering* [3] where a single universe, i.e. view, suffices for learning but the aim is on binding different views to improve the classification accuracy and/or accelerating the learning process.

As it often causes confusion, we want to emphasize the difference of the concept of parallel universes to feature selection methods [12], feature transformation (such as principle

component analysis and singular value decomposition), and subspace clustering [13,8], whose problem definitions sound similar at first but are very different from what we discuss here. Feature selection methods attempt to discover attributes in a data set that are most relevant to the task at hand. Subspace clustering is an extension of feature selection that seeks to identify different subspaces, i.e. subsets of input features, for the same dataset. These algorithms become particularly useful when dealing with high-dimensional data, where often, many dimensions are irrelevant and can mask existing clusters in noise. The main goal of such algorithms is therefore to uncover subsets of attributes (subspaces), on which subsets of the data are self-similar, i.e. build subspace-clusters, whereas the clustering in parallel universes is given the definition of semantically meaningful universes along with representations of all data in them and the goal is to exploit this information. The objective for our problem definition is to identify clusters located in different universes whereby each cluster models a subset of the data based on some underlying property.

Since standard clustering techniques are not able to cope with parallel universes, one could either restrict the analysis to a single universe at a time or define a descriptor space comprising all universes. However, using only one particular universe omits information encoded in the other representations and the construction of a joint feature space and the derivation of an appropriate distance measure are cumbersome and require great care as it can introduce artifacts or hide and lose clusters that were apparent in a single universe.

3. Related work

Clustering in parallel universes is a relatively new field of research and was first mentioned in [14]. In [11], the DBSCAN algorithm is extended and applied to parallel universes. DBSCAN uses the notion of dense regions by means of core objects, i.e. objects that have a minimum number k of objects in their (ϵ-) neighborhood. A cluster is then defined as a set of (connected) dense regions. The authors extend this concept in two different ways: they define an object as a neighbor of a core object if it is in the ϵ-neighborhood of this core object either (1) in any of the representations or (2) in all of them. The cluster size is finally determined by appropriate values of ϵ and k. Case (1) seems rather weak, having objects in one cluster even though they might not be similar in any of the representational feature spaces. Case (2), in comparison, is very conservative since it does not reveal local clusters, i.e. subsets of the data that only group in a single universe. However, the results in [11] are promising.

Another clustering scheme called "collaborative fuzzy clustering" is based on the FCM algorithm and was introduced in [15]. The author proposes an architecture in which objects described in parallel universes can be processed together with the objective of finding structures that are common to all universes. Clustering is carried out by applying the c-Means algorithm to all universes individually and then by exchanging information from the local clustering results based on the partitioning matrices. Note, the objective function, as introduced in [15], assumes the same number of clusters in each universe and, moreover, a global order on the clusters, which is very restrictive due to the random initialization of FCM.

A supervised clustering technique for parallel universes was given in [14]. It focuses on a model for a particular (minor) class of interest by constructing local neighborhood

histograms, so-called Neighborgrams for each object of interest in each universe. The algorithm assigns a quality value to each Neighborgram and greedily includes the best Neighborgram, no matter from which universe it stems, in the global prediction model. Objects that are covered by this Neighborgram are finally removed from consideration in a sequential covering manner. This process is repeated until the global model has sufficient predictive power. Although the algorithm is powerful to model a minority class, it suffers from computational complexity on larger data sets.

Blum and Mitchell [4] introduced co-training as a semi-supervised procedure whereby two different hypotheses are trained on two distinct representations and then bootstrap each other. In particular they consider the problem of classifying web pages based on the document itself and on anchor texts of inbound hyperlinks. They require a conditional independence of both universes and state that each representation should suffice for learning if enough labeled data were available. The benefit of their strategy is that (inexpensive) unlabeled data augment the (expensive) labeled data by using the prediction in one universe to support the decision making in the other.

Other related work includes reinforcement clustering [18] and extensions of partitioning methods—such as k-Means, k-Medoids, and EM—and hierarchical, agglomerative methods, all in [3].

4. Objective functions

In this section, we introduce all necessary notation, review the FCM [1,7] algorithm and formulate two new objective functions that are suitable to be used for parallel universes. The first one is a generic function that, similar to the standard FCM, has no explicit noise handling and therefore forces a cluster membership prediction for each pattern while the second objective function also incorporates noise detection and, hence, allows patterns to not participate in any cluster. The technical details, i.e. the derivation of the objective functions, can be found in Appendix A.

In the following, we consider U, $1 \leqslant u \leqslant U$, parallel universes, each having representational feature vectors for all objects $\vec{x}_i^{(u)} = (x_{i,1}^{(u)}, \ldots, x_{i,a}^{(u)}, \ldots x_{i,A_u}^{(u)})$ with A_u indicating the dimensionality of the uth universe. We depict the overall number of objects as $|T|$, $1 \leqslant i \leqslant |T|$. We are interested in identifying K_u clusters in universe u. We further assume appropriate definitions of distance functions for each universe $d^{(u)}(\vec{w}_k^{(u)}, \vec{x}_i^{(u)})^2$ where $\vec{w}_k^{(u)} = (\vec{w}_{k,1}^{(u)}, \ldots, \vec{w}_{k,a}^{(u)}, \ldots \vec{w}_{k,A_u}^{(u)})$ denotes the kth prototype in the uth universe.

We confine ourselves to the Euclidean distance in the following. In general, there are no restrictions to the distance metrics other than differentiability. In particular, they do not need to be of the same type in all universes. This is important to note, since we can use the proposed algorithm in the same feature space, i.e. $\vec{x}_i^{(u_1)} = \vec{x}_i^{(u_2)}$ for some u_1 and u_2, but different distance measures in these universes.

4.1. Objective function with no noise detection

The standard FCM algorithm relies on one feature space only and minimizes the accumulated sum of distances between patterns $\vec{x}_i$ and cluster centers $\vec{w}_k$, weighted by the degree of membership to which a pattern belongs to a cluster. Note that we omit the subscript u here, as we consider only one universe

$$J_m = \sum_{i=1}^{|T|} \sum_{k=1}^{K} v_{i,k}^m d(\vec{w}_k, \vec{x}_i)^2. \tag{1}$$

The coefficient $m \in (1, \infty)$ is a fuzzyfication parameter, and $v_{i,k}$ the respective value from the partition matrix, i.e. the degree to which pattern $\vec{x}_i$ belongs to cluster k.

This function is subject to minimization under the constraint

$$\forall i: \quad \sum_{k=1}^{K} v_{i,k} = 1 \tag{2}$$

requiring that the coverage of any pattern i needs to accumulate to 1.

The above objective function assumes all cluster candidates to be located in the same feature space and is therefore not directly applicable to parallel universes. To overcome this, we introduce a matrix $(z_{i,u})_{1 \leqslant i \leqslant |T|, 1 \leqslant u \leqslant U}$ encoding the membership of patterns to universes. A value $z_{i,u}$ close to 1 denotes a strong contribution of pattern $\vec{x}_i$ to the clustering in universe u, and a smaller value, a respectively lesser degree.

The new objective function is given by

$$J_{m,m'} = \sum_{i=1}^{|T|} \sum_{u=1}^{U} (z_{i,u})^{m'} \sum_{k=1}^{K_u} \left(v_{i,k}^{(u)}\right)^m d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2. \tag{3}$$

Parameter $m' \in (1, \infty)$ controls (analogous to m) the fuzzyfication of $z_{i,u}$: the larger m', the more equal the distribution of $z_{i,u}$, giving each pattern an equal impact to all universes. A value close to 1 will strengthen the composition of $z_{i,u}$ and assign high values to universes where a pattern shows good clustering behavior and small values to those where it does not. Note, we now have U, $1 \leqslant u \leqslant U$, different partition matrices $\left(v_{i,k}^{(u)}\right)_{1 \leqslant i \leqslant |T|, 1 \leqslant k \leqslant K_u}$ to assign membership degrees of objects to cluster prototypes in each universe.

As in the standard FCM algorithm, the objective function has to fulfill side constraints. The coverage of a pattern among the partitions in each universe must accumulate to 1:

$$\forall i, u: \sum_{k=1}^{K_u} v_{i,k}^{(u)} = 1. \tag{4}$$

This is similar to the constraint of the single universe FCM in (2) and is required for each universe individually.

Additionally, the membership of a pattern to different universes $z_{i,u}$ has to satisfy standard requirements for membership degrees: it must accumulate to 1 for each object considering all universes and must be in the unit interval, i.e.

$$\forall i: \quad \sum_{u=1}^{U} z_{i,u} = 1. \tag{5}$$

The minimization is done with respect to the parameters $v_{i,k}^{(u)}$, $z_{i,u}$, and $\vec{w}_k^{(u)}$. The derivation of objective function (3) can be found in Appendix A, the final update equations are given by (A.12), (A.7) and (A.14).

4.2. Objective function with noise detection

The objective function as introduced in the previous section has one major drawback: patterns that do not contribute to any of the clusters in any universe still have a great

impact on the cluster formation as the cluster memberships for each individual pattern need to sum up to one. This is not advantageous since data sets in many real world applications, if not all, contain outliers or noisy patterns. Particularly in the presented application domain it may happen that certain structural properties of the data are not captured by any of the given (semantically meaningful!) universes and therefore this portion of the data appears to be noise. The identification of these patterns is important for two reasons: first, as noted above, these patterns influence the cluster formation and can lead to distorted clusters. Secondly, noise patterns may lead to insights on which properties of the underlying data are not well modeled by any of the universe definitions and therefore give hints as to what needs to be addressed when defining new universes or similarity measures.

In order to incorporate noise detection we need to extend our objective function such that it also allows the explicit notion of noise. We adopt an extension introduced by Davé [7], which works on the single universe FCM. The objective function according to Davé is given by

$$J_m = \sum_{i=1}^{|T|} \sum_{k=1}^{K} v_{i,k}^m d(\vec{w}_k, \vec{x}_i)^2 + d_{\text{noise}} \sum_{i=1}^{|T|} \left(1 - \sum_{k=1}^{K} v_{i,k}\right)^m. \tag{6}$$

This equation is similar to (1) except for the last term. It serves as a noise cluster; all objects have a fixed, user-defined distance d_{noise} to this noise cluster. Objects that are not close to any cluster center $\vec{w}_k$ can therefore be detected as noise. The constraint (2) must be softened

$$\forall i: \quad \sum_{k=1}^{K} v_{i,k} \leqslant 1 \tag{7}$$

requiring that the coverage of any pattern i needs to accumulate to 1 at most (the remainder to 1 represents the membership to the noise cluster).

Similar to the last term in (6), we add a new term to our new objective function (3) whose role is to "localize" the noise and place it in a single auxiliary universe

$$J_{m,m'} = \sum_{i=1}^{|T|} \sum_{u=1}^{U} (z_{i,u})^{m'} \sum_{k=1}^{K_u} \left(v_{i,k}^{(u)}\right)^m d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2 + d_{\text{noise}} \sum_{i=1}^{|T|} \left(1 - \sum_{u=1}^{U} z_{i,u}\right)^{m'}. \tag{8}$$

By assigning patterns to this noise universe, we declare them to be outliers in the data set. The parameter d_{noise} reflects the fixed distance between a virtual cluster in the noise universe and all data points. Hence, if the minimum distance between a data point and any cluster in one of the universes is greater than d_{noise}, the pattern is labeled as noise.

The optimization splits into three parts: optimization of the partition values $v_{i,k}^{(u)}$ for each universe; determining the membership degrees of patterns to universes $z_{i,u}$ and finally the adaption of the center vectors of the cluster representatives $\vec{w}_k^{(u)}$.

The update equations of these parameters are given as follows. For the partition values $v_{i,k}$, we get

$$v_{i,k}^{(u)} = \frac{1}{\sum_{\bar{k}=1}^{K_u} \left(\frac{d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2}{d^{(u)}\left(\vec{w}_{\bar{k}}^{(u)}, \vec{x}_i^{(u)}\right)^2} \right)^{\frac{1}{m-1}}}. \tag{9}$$

Note, this equation is independent of the values $z_{i,u}$ and is therefore identical to the update expression in the single universe FCM. The optimization with respect to $z_{i,u}$ yields

$$z_{i,u} = \frac{1}{\sum_{\breve{u}=1}^{U} \left(\frac{\sum_{k=1}^{K_u} \left(v_{i,k}^{(u)}\right)^m d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2}{\sum_{k=1}^{K_{\breve{u}}} \left(v_{i,k}^{(\breve{u})}\right)^m d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2 + d_{\text{noise}}} \right)^{\frac{1}{m'-1}}} \tag{10}$$

and finally the update equation for the adaption of the prototype vectors $\vec{w}_k^{(u)}$ is of the form

$$\vec{w}_k^{(u)} = \frac{\sum_{i=1}^{|T|} \left(z_{i,u}\right)^{m'} \left(v_{i,k}^{(u)}\right)^m \vec{x}_i^{(u)}}{\sum_{i=1}^{|T|} \left(z_{i,u}\right)^{m'} \left(v_{i,k}^{(u)}\right)^m}. \tag{11}$$

Thus, the update of the prototypes depends not only on the partitioning value $v_{i,k}^{(u)}$, i.e. the degree to which pattern i belongs to cluster k in universe u, but also to $z_{i,u}$ representing the membership degrees of patterns to the current universe of interest. Patterns with larger values $z_{i,u}$ will contribute more to the adaption of the prototype vectors, while patterns with a smaller degree accordingly to a lesser extent.

Equipped with these update equations, we can introduce the overall clustering scheme in the next section.

5. Clustering algorithm

Similar to the standard FCM algorithm, clustering is carried out in an iterative manner, involving three steps:

(1) Update of the partition matrices $\left(v_{i,k}^{(u)}\right)$.
(2) Update of the membership degrees $(z_{i,u})$.
(3) Update of the prototypes $(\vec{w}_k^{(u)})$.

More precisely, the clustering procedure is given as:

(1) *Given:* Input pattern set described in U parallel universes: $\vec{x}_i^{(u)}$, $1 \leqslant i \leqslant |T|$, $1 \leqslant u \leqslant U$.
(2) *Select:* A set of distance metrics $d^{(u)}(\cdot,\cdot)^2$, and the number of clusters for each universe K_u, $1 \leqslant u \leqslant U$, define parameter m and m'.
(3) *Initialize:* Partition parameters $v_{i,k}^{(u)}$ with random values and the cluster prototypes by drawing samples from the data. Assign equal weights to all membership degrees $z_{i,u} = \frac{1}{U}$.
(4) *Train:*
(5) *Repeat*
(6) Update partitioning values $v_{i,k}^{(u)}$ according to (9)
(7) Update membership degrees $z_{i,u}$ according to (10)
(8) Compute prototypes $\vec{w}_i^{(u)}$ using (11)
(9) *until* a termination criterion has been satisfied.

The algorithm starts with a given set of universe definitions and the specification of the distance metrics to be used. Also, the number of clusters in each universe needs to be defined in advance. The membership degrees $z_{i,u}$ are initialized with equal weight (line

(3)), thus having the same impact on all universes. The optimization phase in line (5)–(9) is—in comparison to the standard FCM algorithm—extended by the optimization of the universe membership degrees, line (7). The possibilities for the termination criterion in line (9) are manifold, as is also the case in the standard FCM. One can stop after a certain number of iterations or use the change of the value of the objective function (3) between two successive iterations as stopping criteria. There are also more sophisticated approaches, for instance the change to the partition matrices during optimization.

Just like the FCM algorithm, this method suffers from the fact that the user has to specify the number of prototypes to be found. Furthermore, our approach even requires the definition of cluster counts *per* universe. There are numerous approaches to suggest the number of clusters in the case of the standard FCM [19,17,2], to name but a few. Although we have not yet studied their applicability to our problem definition we do believe that some of them can be adapted naturally to be used in our context as well.

6. Experimental results

In order to demonstrate the proposed approach, we generated synthetic data sets with different numbers of parallel universes. For simplicity and in order to visualize the results we restricted the size of a universe to two dimensions and generated two Gaussian distributed clusters per universe. We used 1400 patterns to build groupings by assigning each object to one of the universes and drawing its features in that universe according to the distribution of the cluster (randomly picking one of the two). The features of that object in the other universes were drawn from a uniform distribution, i.e. they likely represent noise in these universes (unless they fall, by chance, into one of the clusters). Fig. 1 shows an example data set with three universes. The top figures show only the objects that were

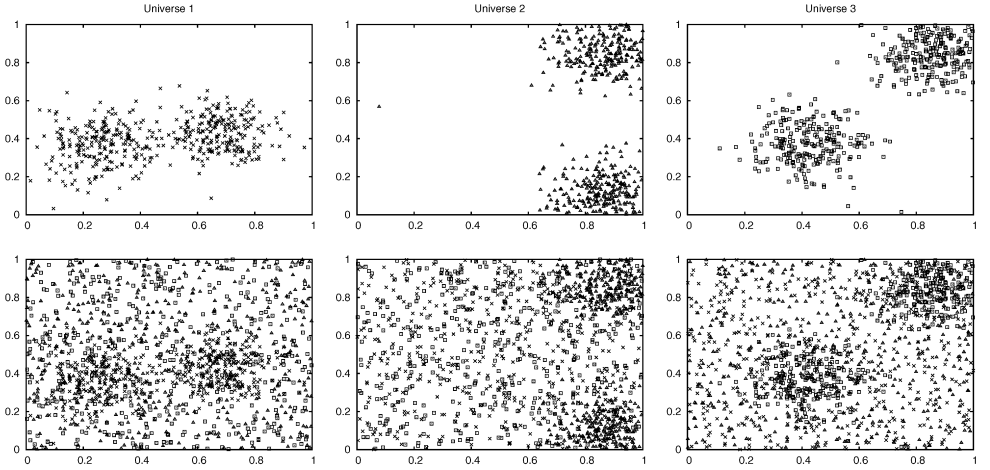

Fig. 1. Three universes of a synthetic data set. The top figures show only objects that were generated within the respective universe (using two clusters per universe). The bottom figures show all patterns; note that most of them (i.e. the ones from the other two universes) are noise in this particular universe. For clarification we use different shapes for objects that originate from different universes.

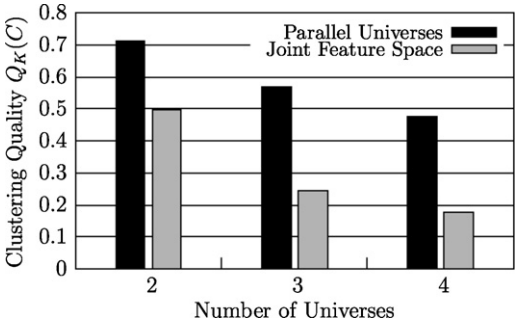

Fig. 2. Clustering quality for 3 different data sets. The number of universes ranges from 2 to 4 universes. Note how the cluster quality of the joint feature space drops sharply whereas the parallel universe approach seems less affected. An overall decline of cluster quality is to be expected since the number of clusters to be detected increases.

generated to cluster in the respective universe. The bottom figures show all patterns, i.e. also the patterns that cluster in the other universes. They define the reference clustering. In this example, when looking solely at one universe, about 2/3 of the data does not contribute to clustering and therefore are noise in that universe.

To compare the results we applied the FCM algorithm [1] to the joint feature space of all universes and set the number of desired clusters to the overall number of generated clusters.

The cluster membership decision for the single-universe FCM is based on the highest value of the partition values, i.e. the cluster to a pattern i is determined by $\bar{k} = \arg\max_{1 \leqslant k \leqslant K}\{v_{i,k}\}$.

When the universe information is taken into account, a cluster decision is based on the memberships to universes $z_{i,u}$ and memberships to clusters $v_{i,k}^{(u)}$. The "winning" universe is determined by $\bar{u} = \arg\max_{1 \leqslant u \leqslant U}\{z_{i,u}\}$ and the corresponding cluster in $\bar{u}$ is calculated as $\bar{k} = \arg\max_{1 \leqslant k \leqslant K_{\bar{u}}}\{v_{i,k}^{(\bar{u})}\}$.

We used the following quality measure to evaluate the clustering outcome and compare it to the reference clustering [11]

$$Q_K(C) = \sum_{C_i \in C} \frac{|C_i|}{|T|} \cdot (1 - \text{entropy}_K(C_i)),$$

where K is the reference clustering, i.e. the clusters as generated, C the clustering to evaluate, and $\text{entropy}_K(C_i)$ the entropy of cluster C_i with respect to K. This function is 1 if C equals K and 0 if all clusters are completely mixed such that they all contain an equal fraction of the clusters in K or all points are predicted to be noise. Thus, the higher the value, the better the clustering.

Fig. 2 summarizes the quality values for 3 experiments. The number of universes ranges from 2 to 4. The left bar for each experiment in Fig. 2 shows the quality value when using the new objective function as introduced in Section 4.1, i.e. with incorporating the knowledge of parallel universes but no explicit noise detection. The right bar shows the quality value when applying the standard FCM to the joint feature space. Clearly, for this data set, our algorithm takes advantage of the information encoded in different universes and identifies the major parts of the original clusters much better.

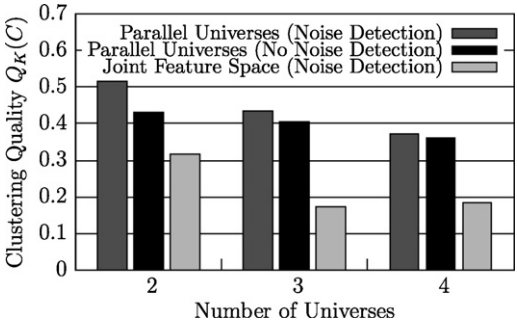

Fig. 3. Results on the artificial dataset with 600 patterns being noise, i.e. not contributing to any cluster. When using our new algorithms (the two left bars for each experiment) the quality values are always greater than the value for the FCM with noise cluster [7] applied to the joint feature space.

In a second experiment, we artificially added 600 noise patterns[1] in order to test the ability of noise detection. The patterns' features were drawn from a random distribution in all universes, hence, they likely represent noise. We then applied our new algorithm in parallel universes with and without noise detection and compared the results to the extended FCM algorithm with noise detection [7] applied to the joint feature space. The crisp cluster membership was based on the degree of membership of a pattern to the auxiliary noise cluster: whenever this value was higher than the maximum membership to any of the real clusters, the pattern was labeled as noise, i.e. $\max_{1\leqslant k\leqslant K}\{v_{i,k}\} < 1 - \sum_{k=1}^{K} v_{i,k}$. Similarly, in the case of the algorithm in parallel universes, a pattern is detected as noise when the degree of membership to the auxiliary noise universe is higher than to any real universe, $\max_{1\leqslant u\leqslant U}\{z_{i,u}\} < 1 - \sum_{u=1}^{U} z_{i,u}$.

Fig. 3 summarizes the quality values for this experiment. Clearly, when allowing the algorithm to label patterns as noise, the quality value increases. However, when applying FCM to the joint feature space (right most bar), most of the data was labeled as noise. It was noticeable, that the noise detection (30% of the data was generated randomly such that it should not cluster in any universe) decreased when there were more universes, since the number of clusters—and therefore the chance to "hit" one of them when drawing the features of a noise object—increased for this artificial data. As a result, the difference in quality between the clustering algorithm, which allows noise detection, and the clustering algorithm that forces a cluster prediction declines when there are more universes. This effect occurs no matter how carefully the noise distance parameter d_{noise} is chosen.

However, if we have only few universes, the difference is quite obvious. Fig. 4 visually demonstrates the clusters from the foregoing example as they are determined by the fuzzy c-Means algorithm in parallel universes: the top figures show the outcome when using the objective function introduced in Section 4.1, i.e. without noise detection, and the bottom figures show the clusters when allowing noise detection (Section 4.2). The figures show only the patterns that are part of clusters in the respective universe; other patterns, either covered by clusters in the remaining universes or detected as noise, are filtered out. Note how the clusters in the top figures are spread and contain patterns that obviously do not make much sense for this clustering. This is due to the fact that the algorithm is not

[1] The overall number of patterns is therefore 2000 patterns.

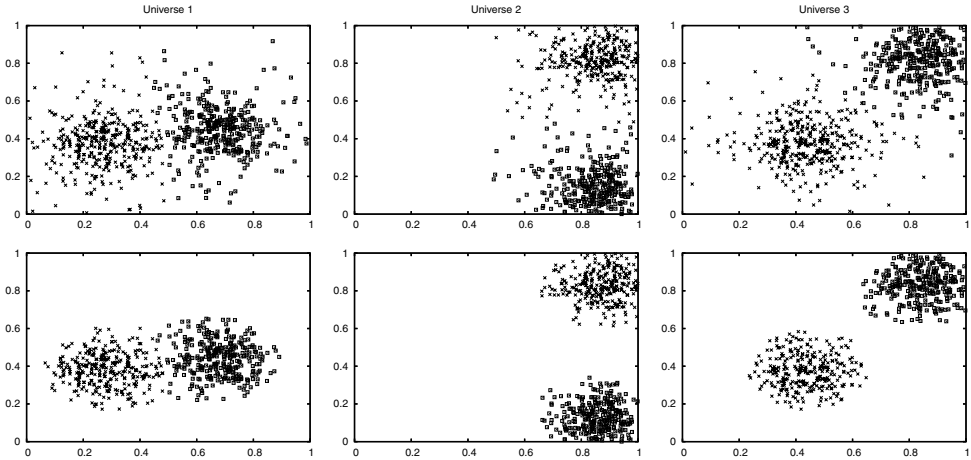

Fig. 4. The top figures show the clusters as they are found when applying the algorithm with no noise detection. The bottom figures show the clusters found by the algorithm using noise detection. While the clusters in the top figures contain patterns that do not appear natural for this clustering, the clustering with noise detection reveals those patterns and builds up clear groupings.

allowed to discard such patterns as noise: each pattern must be assigned to a cluster. The bottom figures, in comparison, show the clusters as well-shaped, dense regions. Patterns that in the top figures distort the clusters are not included here. It shows nicely that the algorithm does not force a cluster prediction and will recognize these patterns as noise.

We chose this kind of data generation to test the ability to detect clusters that are blurred by noise. Particularly in biological data analysis it is common to have noisy data for which different descriptors are available and each by itself exhibits only little clustering power.

7. Conclusion

We considered the problem of unsupervised clustering in parallel universes, i.e. problems where multiple representations are available for each object. We developed an extension of the fuzzy c-Means algorithm with noise detection that uses membership degrees to model the impact of objects to the clustering in a particular universe. By incorporating these membership values into the objective function, we were able to derive update equations which minimize the objective function with respect to these values, the partition matrices, and the prototype center vectors. In order to model the concept of noise, i.e. patterns that apparently are not contained in any of the clusters in any universe, we introduced an auxiliary noise universe that has one single cluster to which all objects have a fixed, pre-defined distance. Patterns that are not covered by any of the clusters are assigned a high membership to this universe and can therefore be revealed as noise.

The clustering algorithm itself works in an iterative manner similar to the standard FCM using the above update equations to compute a (local) minimum. The result is clusters located in different parallel universes, each modeling only a subset of the overall data and ignoring data that do not contribute to clustering in a universe.

We demonstrated that the algorithm performs well on a synthetic data set and nicely exploits the information of having different universes.

Further studies will concentrate on the overlap of clusters. The proposed objective function rewards clusters that only occur in one universe. Objects that cluster well in more than one universe could possibly be identified when having balanced membership values to the universes but very unbalanced partitioning values for the cluster memberships within these particular universes.

Other studies will continue to focus on the applicability of the proposed method to real world data and heuristics that adjust the number of clusters per universe.

Acknowledgement

This work was partially supported by DFG Research Training Group GK-1042 "Explorative Analysis and Visualization of Large Information Spaces".

Appendix A

In order to compute a minimum of the objective function (3) with respect to (4) and (5), we exploit a Lagrange technique to merge the constrained part of the optimization problem with the unconstrained one. As before, we use u, $1 \leqslant u \leqslant U$ as universe count, whereby each universe comprises representational feature vectors for all objects $\vec{x}_i^{(u)} = (x_{i,1}^{(u)}, \ldots, x_{i,a}^{(u)}, \ldots x_{i,A_u}^{(u)})$ with A_u indicating the dimensionality of the uth universe. The number of objects is depicted as $|T|$, $1 \leqslant i \leqslant |T|$ and the number of clusters in universe u as K_u. Appropriate definitions of distance functions for each universe $d^{(u)}(\vec{w}_k^{(u)}, \vec{x}_i^{(u)})$ are assumed to be given where $\vec{w}_k^{(u)} = (\vec{w}_{k,1}^{(u)}, \ldots, \vec{w}_{k,a}^{(u)}, \ldots \vec{w}_{k,A_u}^{(u)})$ denotes the kth prototype in the uth universe.

Note, we skip the extra notation of the noise universe in (8); it can be seen as an additional universe, i.e. the number of universes is $U+1$, that has one cluster to which all patterns have a fixed distance of d_{noise}. The derivation can then be applied as follows.

It leads to a new objective function F_i:

$$F_i = \sum_{u=1}^{U} (z_{i,u})^{m'} \sum_{k=1}^{K_u} \left(v_{i,k}^{(u)}\right)^m d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2 + \sum_{u=1}^{U} \lambda'_u \left(1 - \sum_{k=1}^{K_u} v_{i,k}^{(u)}\right) + \lambda \left(1 - \sum_{u=1}^{U} z_{i,u}\right), \tag{A.1}$$

which we minimize individually for each pattern $\vec{x}_i$. The parameters λ and λ'_u, $1 \leqslant u \leqslant U$, denote the Lagrange multipliers to take (4) and (5) into account. The necessary conditions leading to local minima of F_i read as

$$\frac{\partial F_i}{\partial z_{i,u}} = 0, \quad \frac{\partial F_i}{\partial v_{i,k}^{(u)}} = 0, \quad \frac{\partial F_i}{\partial \lambda} = 0, \quad \frac{\partial F_i}{\partial \lambda'_u} = 0, \qquad 1 \leqslant u \leqslant U, \quad 1 \leqslant k \leqslant K_u. \tag{A.2}$$

In the following we will derive update equations for the z and v parameters. Evaluating the first derivative of the equations in (A.2) yields the expression

$$\frac{\partial F_i}{\partial z_{i,u}} = m'(z_{i,u})^{m'-1} \sum_{k=1}^{K_u} \left(v_{i,k}^{(u)}\right)^m d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2 - \lambda = 0$$

and hence

$$z_{i,u} = \left(\frac{\lambda}{m'}\right)^{\frac{1}{m'-1}} \left(\frac{1}{\sum_{k=1}^{K_u}\left(v_{i,k}^{(u)}\right)^m d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2}\right)^{\frac{1}{m'-1}}. \tag{A.3}$$

We can rewrite the above equation

$$\left(\frac{\lambda}{m'}\right)^{\frac{1}{m'-1}} = z_{i,u} \left(\sum_{k=1}^{K_u} \left(v_{i,k}^{(u)}\right)^m d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2\right)^{\frac{1}{m'-1}}. \tag{A.4}$$

From the derivative of F_i w.r.t. λ in (A.2), it follows:

$$\begin{aligned} &\frac{\partial F_i}{\partial \lambda} = 1 - \sum_{u=1}^{U} z_{i,u} = 0, \\ &\sum_{u=1}^{U} z_{i,u} = 1, \end{aligned} \tag{A.5}$$

which returns the normalization condition as in (5). Using the formula for $z_{i,u}$ in (A.3) and integrating it into expression (A.5) we compute

$$\begin{aligned} &\sum_{u=1}^{U} \left(\frac{\lambda}{m'}\right)^{\frac{1}{m'-1}} \left(\frac{1}{\sum_{k=1}^{K_u}\left(v_{i,k}^{(u)}\right)^m d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2}\right)^{\frac{1}{m'-1}} = 1, \\ &\left(\frac{\lambda}{m'}\right)^{\frac{1}{m'-1}} \sum_{u=1}^{U} \left(\frac{1}{\sum_{k=1}^{K_u}\left(v_{i,k}^{(u)}\right)^m d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2}\right)^{\frac{1}{m'-1}} = 1. \end{aligned} \tag{A.6}$$

We make use of (A.4) and substitute $\left(\frac{\lambda}{m'}\right)^{\frac{1}{m'-1}}$ in (A.6). Note, we use $\bar{u}$ as the parameter index of the sum to address the fact that it covers all universes, whereas u denotes the current universe of interest. It follows:

$$1 = z_{i,u} \left(\sum_{k=1}^{K_u} \left(v_{i,k}^{(u)}\right)^m d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2\right)^{\frac{1}{m'-1}} \cdot \sum_{\bar{u}=1}^{U} \left(\frac{1}{\sum_{k=1}^{K_{\bar{u}}}\left(v_{i,k}^{(\bar{u})}\right)^m d^{(\bar{u})}\left(\vec{w}_k^{(\bar{u})}, \vec{x}_i^{(\bar{u})}\right)^2}\right)^{\frac{1}{m'-1}},$$

which can be simplified to

$$1 = z_{i,u} \sum_{\bar{u}=1}^{U} \left(\frac{\sum_{k=1}^{K_u}\left(v_{i,k}^{(u)}\right)^m d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2}{\sum_{k=1}^{K_{\bar{u}}}\left(v_{i,k}^{(\bar{u})}\right)^m d^{(\bar{u})}\left(\vec{w}_k^{(\bar{u})}, \vec{x}_i^{(\bar{u})}\right)^2}\right)^{\frac{1}{m'-1}}$$

and returns an immediate update expression for the membership $z_{i,u}$ of pattern i to universe u

$$z_{i,u} = \frac{1}{\sum_{\bar{u}=1}^{U} \left(\frac{\sum_{k=1}^{K_u} \left(v_{i,k}^{(u)}\right)^m d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2}{\sum_{k=1}^{K_{\bar{u}}} \left(v_{i,k}^{(\bar{u})}\right)^m d^{(\bar{u})}\left(\vec{w}_k^{(\bar{u})}, \vec{x}_i^{(\bar{u})}\right)^2} \right)^{\frac{1}{m'-1}}}. \tag{A.7}$$

Analogous to the calculations above we can derive the update equation for value $v_{i,k}^{(u)}$ which represents the partitioning value of pattern i to cluster k in universe u. From (A.2) it follows:

$$\frac{\partial F_i}{\partial v_{i,k}^{(u)}} = (z_{i,u})^{m'} m \left(v_{i,k}^{(u)}\right)^{m-1} d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2 - \lambda_u' = 0$$

and thus

$$v_{i,k}^{(u)} = \left(\frac{\lambda_u'}{m(z_{i,u})^{m'} d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2} \right)^{\frac{1}{m-1}}, \tag{A.8}$$

$$\left(\frac{\lambda_u'}{m(z_{i,u})^{m'}} \right)^{\frac{1}{m-1}} = v_{i,k}^{(u)} \left(d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2 \right)^{\frac{1}{m-1}}. \tag{A.9}$$

Zeroing the derivative of F_i w.r.t. λ_u' will result in condition (4), ensuring that the partition values sum up to 1, i.e.

$$\frac{\partial F_i}{\partial \lambda_u'} = 1 - \sum_{k=1}^{K_u} v_{i,k}^{(u)} = 0. \tag{A.10}$$

We use (A.8) and (A.10) to come up with

$$\begin{aligned} 1 &= \sum_{k=1}^{K_u} \left(\frac{\lambda_u'}{m(z_{i,u})^{m'} d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2} \right)^{\frac{1}{m-1}}, \\ 1 &= \left(\frac{\lambda_u'}{m(z_{i,u})^{m'}} \right)^{\frac{1}{m-1}} \sum_{k=1}^{K_u} \left(\frac{1}{d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2} \right)^{\frac{1}{m-1}}. \end{aligned} \tag{A.11}$$

Eq. (A.9) allows us to replace the first multiplier in (A.11). We will use the $\bar{k}$ notation to point out that the sum in (A.11) considers all partitions in a universe and k to denote one particular cluster coming from (A.8),

$$\begin{aligned} 1 &= v_{i,k}^{(u)} \left(d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2 \right)^{\frac{1}{m-1}} \cdot \sum_{\bar{k}=1}^{K_u} \left(\frac{1}{d^{(u)}\left(\vec{w}_{\bar{k}}^{(u)}, \vec{x}_i^{(u)}\right)^2} \right)^{\frac{1}{m-1}}, \\ 1 &= v_{i,k}^{(u)} \sum_{\bar{k}=1}^{K_u} \left(\frac{d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2}{d^{(u)}\left(\vec{w}_{\bar{k}}^{(u)}, \vec{x}_i^{(u)}\right)^2} \right)^{\frac{1}{m-1}}. \end{aligned}$$

Finally, the update rule for $v_{i,k}^{(u)}$ arises as

$$v_{i,k}^{(u)} = \frac{1}{\sum_{\bar{k}=1}^{K_u} \left(\frac{d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2}{d^{(u)}\left(\vec{w}_{\bar{k}}^{(u)}, \vec{x}_i^{(u)}\right)^2} \right)^{\frac{1}{m-1}}}. \tag{A.12}$$

For the sake of completeness we also derive the update rules for the cluster prototypes $\vec{w}_k^{(u)}$. We confine ourselves to the Euclidean distance here, assuming the data is normalized[2]:

$$d^{(u)}\left(\vec{w}_k^{(u)}, \vec{x}_i^{(u)}\right)^2 = \sum_{a=1}^{A_u} \left(w_{k,a}^{(u)} - x_{i,a}^{(u)} \right)^2 \tag{A.13}$$

with A_u the number of dimensions in universe u and $w_{k,a}^{(u)}$ the value of the prototype in dimension a. $x_{i,a}^{(u)}$ is the value of the ath attribute of pattern i in universe u, respectively. The necessary condition for a minimum of the objective function (3) is of the form $\nabla_{\vec{w}_k^{(u)}} J = 0$. Using the Euclidean distance as given in (A.13) we obtain

$$\begin{aligned}
&\frac{\partial J_{m,m'}}{\partial w_{k,a}^{(u)}} \stackrel{!}{=} 0, \\
&2 \sum_{i=1}^{|T|} (z_{i,u})^{m'} \left(v_{i,k}^{(u)} \right)^m \left(w_{k,a}^{(u)} - x_{i,a}^{(u)} \right) = 0, \\
&w_{k,a}^{(u)} \sum_{i=1}^{|T|} (z_{i,u})^{m'} \left(v_{i,k}^{(u)} \right)^m = \sum_{i=1}^{|T|} (z_{i,u})^{m'} \left(v_{i,k}^{(u)} \right)^m x_{i,a}^{(u)}, \\
&w_{k,a}^{(u)} = \frac{\sum_{i=1}^{|T|} (z_{i,u})^{m'} \left(v_{i,k}^{(u)} \right)^m x_{i,a}^{(u)}}{\sum_{i=1}^{|T|} (z_{i,u})^{m'} \left(v_{i,k}^{(u)} \right)^m}.
\end{aligned} \tag{A.14}$$

References

[1] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

[2] James C. Bezdek, Richard J. Hathaway, VAT: a tool for visual assessment of (cluster) tendency, in: Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN '02), 2002, pp. 2225–2230.

[3] Steffen Bickel, Tobias Scheffer, Multi-view clustering, in: Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04), 2004, pp. 19–26.

[4] Avrim Blum, Tom Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory (COLT'98), ACM Press, 1998, pp. 92–100.

[5] Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck, Dejan V. Vranić, An experimental effectiveness comparison of methods for 3D similarity search, International Journal on Digital Libraries (Special issue on Multimedia Contents and Management in Digital Libraries) 6 (1) (2006) 39–54.

[6] G. Cruciani, P. Crivori, P.-A. Carrupt, B. Testa, Molecular fields in quantitative structure-permeation relationships: the VolSurf approach, Journal of Molecular Structure 503 (2000) 17–30.

[2] The derivation of the updates using other than the Euclidean distance works in a similar manner.

[7] Rajesh N. Davé, Characterization and detection of noise in clustering, Pattern Recognition Letters 12 (1991) 657–664.
[8] Jerome H. Friedman, Jacqueline J. Meulman, Clustering objects on subsets of attributes, Journal of the Royal Statistical Society 66 (4) (2004).
[9] David J. Hand, Heikki Mannila, Padhraic Smyth, Principles of Data Mining, MIT Press, 2001.
[10] Frank Höppner, Frank Klawoon, Rudolf Kruse, Thomas Runkler, Fuzzy Cluster Analysis, John Wiley, Chichester, England, 1999.
[11] Karin Kailing, Hans-Peter Kriegel, Alexey Pryakhin, Matthias Schubert, Clustering multi-represented objects with noise, in: PAKDD, 2004, pp. 394–403.
[12] Huan Liu, Hiroshi Motoda, Feature Selection for Knowledge Discovery & Data Mining, Kluwer Academic Publishers, 1998.
[13] Lance Parsons, Ehtesham Haque, Huan Liu, Subspace clustering for high dimensional data: a review, SIGKDD Explorations, Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining 6 (1) (2004) 90–105.
[14] David E. Patterson, Michael R. Berthold, Clustering in parallel universes, in: Proceedings of the 2001 IEEE Conference in Systems, Man and Cybernetics, IEEE Press, 2001.
[15] Witold Pedrycz, Collaborative fuzzy clustering, Pattern Recognition Letters 23 (14) (2002) 1675–1686.
[16] Ansgar Schuffenhauer, Valerie J. Gillet, Peter Willett, Similarity searching in files of three-dimensional chemical structures: analysis of the bioster database using two-dimensional fingerprints and molecular field descriptors, Journal of Chemical Information and Computer Sciences 40 (2) (2000) 295–307.
[17] N.B. Venkateswarlu, P.S.V.S.K. Raju, Fast ISODATA clustering algorithms, Pattern Recognition 25 (3) (1992) 335–342.
[18] Jidong Wang, Hua-Jun Zeng, Zheng Chen, Hongjun Lu, Li Tao, Wei-Ying Ma, ReCoM: reinforcement clustering of multi-type interrelated data objects, in: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'03), 2003, pp. 274–281.
[19] R.R. Yager, D.P. Filev, Approximate clustering via the mountain method, IEEE Transactions on Systems, Man and Cybernetics 24 (8) (1994) 1279–1284.