Fuzzy Clustering in Parallel Universes with Noise Detection

Bernd Wiswedel and Michael R. Berthold
Department of Computer and Information Science, University of Konstanz
78457 Konstanz, Germany
{wiswedel,berthold}@inf.uni-konstanz.de

Abstract

We present an extension of the fuzzy c-Means algorithm that operates on different feature spaces, so-called parallel universes, simultaneously 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 likely represent noise. The outcome of the algorithm are 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. In this paper we show how the fuzzy c-Means algorithm can be extended to operate in parallel universes and illustrate the usefulness of this method using results on artificial data sets.

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 [7, 8]. Most of the methods assume the data to be given in a single (mostly high-dimensional numeric) feature space. In some 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 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. [5]. Further similarities involve the comparison of chemical graphs, inter-atomic distances, and molecular field descriptors. However, it has been shown that often a single descriptor fails to show satisfactory prediction results [13].

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]. Parts in CAD-catalogues can be represented by 3D models, polygon meshes or textual descriptions. Image descriptors can rely on textual keywords, color information, or other properties [9].

In the following we denote these multiple representations, i. e. different descriptor spaces, as *Parallel Universes* [11, 16], each of which having 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 will discuss in more detail the concept of parallel universes; section 3 presents related work. We formulate our new clustering scheme in section 4 and illustrate its usefulness with some numeric examples in section 5.

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 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.

Note, the concept of parallel universes is not related to *Subspace Clustering* [10], even though it seems so at first. Subspace clustering algorithms seek to identify different subspaces, i. e. subsets of input features, in a dataset. This becomes 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 clusters and subspaces containing only a small, but dense fraction of the data, whereas the clustering in parallel universes is given the definition of all data in all universes and the goal is to exploit this information.

The objective for our problem definition is on identifying 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.

3 Related Work

Clustering in parallel universes is a relatively new field of research. In [9], 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 through 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 [9] are promising.

Another clustering scheme called "Collaborative fuzzy clustering" is based on the FCM algorithm and was introduced in [12]. 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 [12], 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 [11]. 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.

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 [15] and extensions of partitioning methods—such as k-Means, k-Medoids, and EM—and hierarchical, agglomerative methods, all in [3].

4 Clustering Algorithm

In this section, we introduce all necessary notation, review the FCM [1, 6] algorithm and formulate a new objective function that is suitable to be used for parallel universes. The technical details, i. e. the derivation of the objective function, can be found in the appendix.

In the following, we consider U, $1 \leq u \leq U$, parallel universe, each having representational feature vec-

tors for all objects $\vec{x}_{i,u} = (x_{i,u,1}, \ldots, x_{i,u,a}, \ldots x_{i,u,A_u})$ with A_u the dimensionality of the u-th universe. We depict the overall number of objects as $|T|$, $1 \leq i \leq |T|$. We are interested in identifying c_u clusters in universe u. We further assume appropriate definitions of distance functions for each universe $d_u\left(\vec{w}_{k,u}, \vec{x}_{i,u}\right)^2$ where $\vec{w}_{k,u} = (\vec{w}_{k,u,1}, \ldots, \vec{w}_{k,u,a}, \ldots \vec{w}_{k,u,A_u})$ denotes the k-th prototype in the u-th universe.

We confine ourselves to the Euclidean distance in the following. In general, there are no restrictions to the distance metrics other than the 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 any u_1 and u_2, but different distance measure across the universes.

4.1 Formulation of new objective function

A 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. We refer here to an objective function that also includes noise detection [6]. Note that we omit the subscript u here, as we consider only one universe:

$$J_m = \sum_{i=1}^{|T|} \sum_{k=1}^{c} v_{i,k}^m \, d\left(\vec{w}_k, \vec{x}_i\right)^2 + \delta^2 \sum_{i=1}^{|T|} \left(1 - \sum_{k=1}^{c} v_{i,k}\right)^m . \quad (1)$$

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. The last term serves as a noise cluster; all objects have an fixed, user-defined distance δ^2 to it. Objects that are not close to any cluster center $\vec{w}_k$ are therefore detected as noise.

This function is subject to minimization under the constraint

$$\forall\, i : \sum_{k=1}^{c} v_{i,k} \leq 1\,, \quad (2)$$

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

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 $(z_{i,u})$, $1 \leq i \leq |T|$, $1 \leq u \leq U$, 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. $z_{i,u}$ has to satisfy standard requirements for membership degrees: it must accumulate to at most 1 considering all universes and must be in the unit interval.

The new objective function is given with

$$J_{m,n} = \sum_{i=1}^{|T|} \sum_{u=1}^{U} z_{i,u}^n \sum_{k=1}^{c_u} v_{i,k,u}^m d_u\left(\vec{w}_{k,u}, \vec{x}_{i,u}\right)^2 + \delta^2 \sum_{i=1}^{|T|} \left(1 - \sum_{u=1}^{U} z_{i,u}\right)^n . \quad (3)$$

Parameter $n \in (1, \infty)$ allows (analogous to m) to have impact on the fuzzyfication of $z_{i,u}$: The larger n 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 different partition matrices $(v_{i,k})$ to assign membership degrees of objects to cluster prototypes. Similar to the objective function (1), the last term's role is to "localize" the noise and place it in a single auxiliary universe. By assigning patterns to this noise universe, we declare them to be outliers in the data set. The parameter δ^2 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 becomes greater than δ^2, the pattern is labeled as noise.

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:

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

This is similar to the constraint of the single universe FCM in (2) but requires to a strict sum of 1 since we do not have a noise cluster in each universe.

Additionally, as mentioned above, the membership of a pattern to different universes has to be at most 1, i. e.

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

The remainder to 1 encodes the membership to the noise cluster mentioned above.

The minimization is done with respect to the parameters $v_{i,k,u}$, $z_{i,u}$, and $\vec{w}_{k,u}$. Since the derivation of the objective function is more of technical interest, please refer to the appendix for details.

The optimization splits into three parts. The 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 in (6), (7), and (8). For the partition values $v_{i,k,u}$, it follows

$$v_{i,k,u} = \frac{1}{\sum_{\bar{k}=1}^{c_u} \left(\frac{d_u(\vec{w}_{k,u}, \vec{x}_{i,u})^2}{d_u(\vec{w}_{\bar{k},u}, \vec{x}_{i,u})^2} \right)^{\frac{1}{m-1}}}. \quad (6)$$

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

$$z_{i,u} = \frac{1}{\sum_{\bar{u}=1}^{U} \left(\frac{\sum_{k=1}^{c_u} v_{i,k,u}^m d_u(\vec{w}_{k,u}, \vec{x}_{i,u})^2}{\sum_{k=1}^{c_{\bar{u}}} v_{i,k,\bar{u}}^m d_{\bar{u}}(\vec{w}_{k,\bar{u}}, \vec{x}_{i,\bar{u}})^2 + \delta^2} \right)^{\frac{1}{n-1}}}, \quad (7)$$

and update equation for the adaption of the prototype vectors $\vec{w}_{k,u}$ is of the form

$$w_{k,u,a} = \frac{\sum_{i=1}^{|T|} z_{i,u}^n \, v_{i,k,u}^m \, x_{i,u,a}}{\sum_{i=1}^{|T|} z_{i,u}^n \, v_{i,k,u}^m}. \quad (8)$$

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.

4.2 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 (v)
2. Update of the membership degrees (z)
3. Update of the prototypes ($\vec{w}$)

More precisely, the clustering procedure is given as:

(1) *Given:* Input pattern set described in U parallel universes: $\vec{x}_{i,u}$, $1 \leq i \leq |T|$, $1 \leq u \leq U$
(2) *Select:* A set of distance metrics $d_u(\cdot,\cdot)^2$, and the number of clusters for each universe c_u, $1 \leq u \leq U$, define parameter m and n
(3) *Initialize:* Partition parameters $v_{i,k,u}$ with random values and the cluster prototypes by drawing samples from the data. Assign equal weight to all membership degrees $z_{i,u} = \frac{1}{U}$.
(4) *Train:*
(5) *Repeat*
(6) Update partitioning values $v_{i,k,u}$ according to (6)
(7) Update membership degrees $z_{i,u}$ according to (7)
(8) Compute prototypes $\vec{w}_{i,u}$ using (8)
(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 use. 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) to (9) is—in comparison to the standard FCM algorithm—extended by the optimization of the membership degrees, line (7). The possibilities for the termination criterion in line (9) are manifold. 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 the 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, [17, 14, 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 to be used in our context as well.

5 Experimental Results

In order to demonstrate this approach, we generated synthetic data sets with different numbers of parallel universes. For simplicity we restricted the size of a universe to 2 dimensions and generated 2 Gaussian distributed clusters per universe. We used 70% of the data (overall cardinality 2000 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

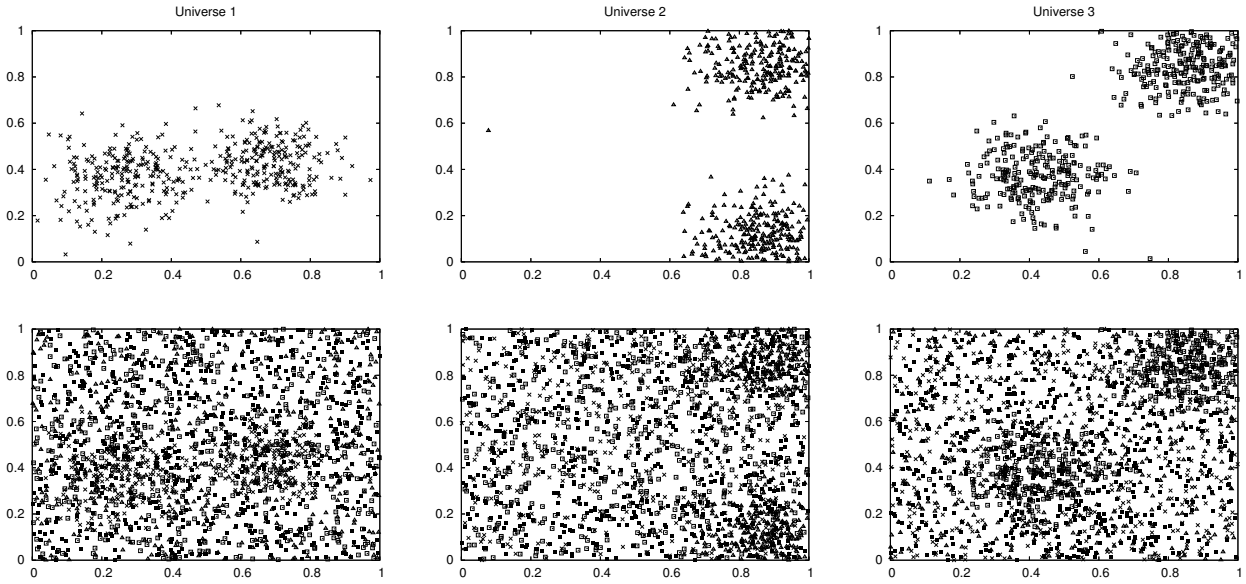

Figure 1. Three universes of a synthetic data set. The top figures show only objects that were generated within the respective universe (using two clusters per universe). The bottom figures show all patterns; note that most of them (i. e. the ones from the other two universes), are noise in this particular universe. They also show the patterns that were not assigned to any of the cluster and represent noise in all of the universes. For clarification we use different shapes for objects that originate from different universes.

universes were drawn from a uniform distribution, i. e. they represent noise in these universes. The remaining 30% of the overall data was generated to be noise in all universes to test the ability of the algorithm to identify patterns that do not cluster at all. Figure 1 shows an example data set with three universes. The top figures show only the objects that were generated to cluster in the respective universe. They define the reference clustering. The bottom figures include all objects, i. e. patterns that cluster in any of the universes plus 30% noise, and show the universes as they are presented to the clustering algorithm. In this example, when looking solely at one universe, about 3/4 of the data does not contribute to clustering and therefore are noise in that universe[1].

To compare the results we applied the FCM algorithm with an auxiliary noise cluster as presented in [6] to the joint feature space of all universes and set the number of desired clusters to the overall number of generated clusters. Thus, the numbers of dimensions and clusters were two times the number of universes. In order to test the ability of noise detection we also applied the fuzzy clustering algorithm for parallel universes without noise universe [16]. The objective function is similar to (3) but with no explicit notion of noise. The algorithm partitions the data such that each pattern is assigned to one of the 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 $\bar{k} = \arg\max_{1\leq k\leq c}\{v_{i,k}\}$. If this value is less than the membership to the noise cluster, $v_{i,\bar{k}} < 1 - \sum_k^{c_u} v_{i,k}$, the pattern is labeled as noise.

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 $\bar{u} = \arg\max_{1\leq u\leq U}\{z_{i,u}\}$. If this value is less than the membership degree to the noise universe, $z_{i,\bar{u}} < 1 - \sum_u^U z_{i,u}$, the pattern is labeled as noise, otherwise the cluster is calculated as $\bar{k} = \arg\max_{1\leq k\leq c_{\bar{u}}}\{v_{i,k,c_{\bar{u}}}\}$.

We used the following quality measure to compare different clustering results [9]:

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

where K is the reference clustering, i. e. the clusters as generated, C the clustering to evaluate, and $\mathrm{entropy}_K(C_i)$ the entropy of cluster C_i with respect to K. This function is 1

[1] More precisely 77% which is 2/3 of 70% clustering in other universes plus 30% overall noise.

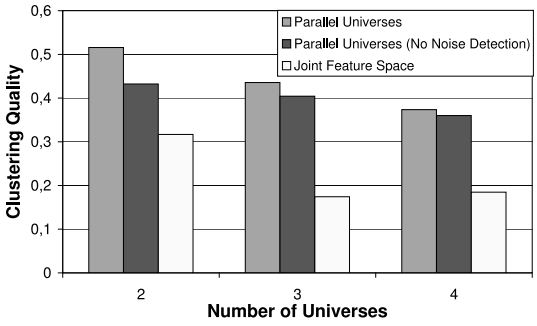

Figure 2. Clustering quality for 3 different data sets. The number of universes ranges from 2 to 4 universes. Note how the cluster quality of the joint feature space drops sharply whereas the parallel universe approach seems less affected. An overall decline of cluster quality is to be expected since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

if C equals K and 0 if all clusters are completely puzzled such that they all contain an equal fraction of the clusters in K or no clusters are detected at all. Thus, the higher the value, the better the clustering.

Figure 2 summarizes the quality values for 3 experiments compared to the FCM with noise detection [6] and the fuzzy clustering in parallel universes with no noise handling [16]. The number of universes ranges from 2 to 4. 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. However, when applying FCM to the joint feature space, most of the data was labeled as noise. It was noticeable, that the noise detection (30% of the data was generated such that it does not cluster in any universe) decreased when having 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 artifical data. As a result, the difference in quality between our new clustering algorithm which allows noise detection and the clustering algorithm that forces a cluster prediction declines when having more universes. This effect occurs no matter how carefully the noise distance parameter δ^2 is chosen.

However, if we have only few universes, the difference is very obvious. Figure 3 visually demonstrates the clusters from the foregoing example as they are determined by the fuzzy c-Means algorithm in parallel universes (the three top figures) and our new algorithm, i. e. with noise detection (bottom figures). The figures show only the patterns that build 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 allowed to declare such patterns as outliers: each pattern must be assigned to a cluster. The bottom figures, in comparison, show the clusters as round-shaped, dense regions. They have been generated using the new objective function. 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 being 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. Obviously this is by no means proof that the method will always detect clusters spread out over parallel universes but these early results are quite promising.

6 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 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 cluster, 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 will get assigned a high membership to this universe and can therefore be revealed as noise.

The clustering algorithm itself works in an iterative manner using the above update equations to compute a (local) minimum. The result are 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

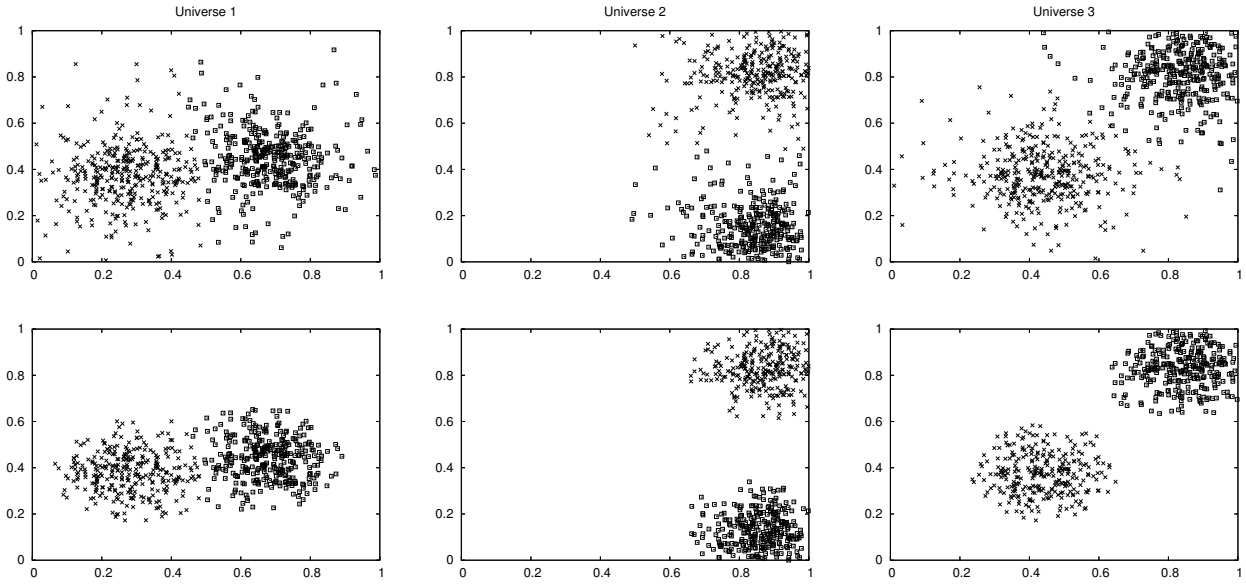

Figure 3. The top figures show the clusters as they are found when applying the algorithm with no noise detection [16]. The bottom figures show the clusters found by the algorithm using noise detection. While the clusters in the top figures contain patterns that do not appear natural for this clustering, the clustering with noise detection reveals those patterns and builds up clear groupings.

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.

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

Acknowledgment

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

Appendix

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. Note we skip the extra notation of the noise universe in (3) as one can think of an additional universe, i. e. the number of universe is $U+1$, that has one cluster to which all patterns have a fixed distance of δ^2. The derivation can then be applied as follows.

It leads to a new objective function F_i that we minimize for each pattern $\vec{x}_i$ individually,

$$F_i = \sum_{u=1}^{U} z_{i,u}^n \sum_{k=1}^{c_u} v_{i,k,u}^m d_u\left(\vec{w}_{k,u}, \vec{x}_{i,u}\right)^2 + \sum_{u=1}^{U} \mu_u \left(1 - \sum_{k=1}^{c_u} v_{i,k,u}\right) + \lambda \left(1 - \sum_{u=1}^{U} z_{i,u}\right). \quad (9)$$

The parameters λ and μ_u, $1 \le u \le U$, denote the Lagrange multiplier to take (4) and (5) into account. The necessary conditions leading to local minima of F_i read as

$$\frac{\partial F_i}{\partial z_{i,u}} = 0, \quad \frac{\partial F_i}{\partial v_{i,k,u}} = 0, \quad \frac{\partial F_i}{\partial \lambda} = 0, \quad \frac{\partial F_i}{\partial \mu_u} = 0\,, \quad 1 \le u \le U, \quad 1 \le k \le c_u. \quad (10)$$

In the following we will derive update equations for the z and v parameters. Evaluating the first derivative of the equations in (10) yields the expression

$$\frac{\partial F_i}{\partial z_{i,u}} = n\, z_{i,u}^{n-1} \sum_{k=1}^{c_u} v_{i,k,u}^m d_u\left(\vec{w}_{k,u}, \vec{x}_{i,u}\right)^2 - \lambda = 0,$$

and hence

$$z_{i,u} = \left(\frac{\lambda}{n}\right)^{\frac{1}{n-1}} \left(\frac{1}{\sum_{k=1}^{c_u} v_{i,k,u}^m d_u\left(\vec{w}_{k,u}, \vec{x}_{i,u}\right)^2}\right)^{\frac{1}{n-1}}. \quad (11)$$

We can rewrite the above equation

$$\left(\frac{\lambda}{n}\right)^{\frac{1}{n-1}} = z_{i,u}\left(\sum_{k=1}^{c_u} v_{i,k,u}^m d_u\left(\vec{w}_{k,u},\vec{x}_{i,u}\right)^2\right)^{\frac{1}{n-1}}. \quad (12)$$

From the derivative of F_i w. r. t. λ in (10), it follows

$$\begin{aligned}\frac{\partial F_i}{\partial \lambda} &= 1 - \sum_{u=1}^{U} z_{i,u} = 0 \\ \sum_{u=1}^{U} z_{i,u} &= 1\,, \end{aligned} \quad (13)$$

which returns the normalization condition as in (5). Using the formula for $z_{i,u}$ in (11) and integrating it into expression (13) we compute

$$\sum_{u=1}^{U}\left(\frac{\lambda}{n}\right)^{\frac{1}{n-1}}\left(\frac{1}{\sum_{k=1}^{c_u} v_{i,k,u}^m d_u\left(\vec{w}_{k,u},\vec{x}_{i,u}\right)^2}\right)^{\frac{1}{n-1}} = 1$$

$$\left(\frac{\lambda}{n}\right)^{\frac{1}{n-1}}\sum_{u=1}^{U}\left(\frac{1}{\sum_{k=1}^{c_u} v_{i,k,u}^m d_u\left(\vec{w}_{k,u},\vec{x}_{i,u}\right)^2}\right)^{\frac{1}{n-1}} = 1 \quad (14)$$

We make use of (12) and substitute $\left(\frac{\lambda}{n}\right)^{\frac{1}{n-1}}$ in (14). Note, we use $\bar{u}$ as parameter index of the sum to address the fact that it covers all universes, whereas u denotes the current universe of interest. It follows

$$\begin{aligned} 1 &= z_{i,u}\left(\sum_{k=1}^{c_u} v_{i,k,u}^m d_u\left(\vec{w}_{k,u},\vec{x}_{i,u}\right)^2\right)^{\frac{1}{n-1}} \\ &\quad \times \sum_{\bar{u}=1}^{U}\left(\frac{1}{\sum_{k=1}^{c_{\bar{u}}} v_{i,k,\bar{u}}^m d_{\bar{u}}\left(\vec{w}_{k,\bar{u}},\vec{x}_{i,\bar{u}}\right)^2}\right)^{\frac{1}{n-1}}, \end{aligned}$$

which can be simplified to

$$1 = z_{i,u}\sum_{\bar{u}=1}^{U}\left(\frac{\sum_{k=1}^{c_u} v_{i,k,u}^m d_u\left(\vec{w}_{k,u},\vec{x}_{i,u}\right)^2}{\sum_{k=1}^{c_{\bar{u}}} v_{i,k,\bar{u}}^m d_{\bar{u}}\left(\vec{w}_{k,\bar{u}},\vec{x}_{i,\bar{u}}\right)^2}\right)^{\frac{1}{n-1}},$$

and returns an immediate update expression for the membership $z_{i,u}$ of pattern i to universe u (see also (7)):

$$z_{i,u} = \frac{1}{\sum_{\bar{u}=1}^{U}\left(\frac{\sum_{k=1}^{c_u} v_{i,k,u}^m d_u(\vec{w}_{k,u},\vec{x}_{i,u})^2}{\sum_{k=1}^{c_{\bar{u}}} v_{i,k,\bar{u}}^m d_{\bar{u}}(\vec{w}_{k,\bar{u}},\vec{x}_{i,\bar{u}})^2}\right)^{\frac{1}{n-1}}}.$$

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 (10) it follows

$$\frac{\partial F_i}{\partial v_{i,k,u}} = z_{i,u}^n\, m\, v_{i,k,u}^{m-1} d_u\left(\vec{w}_{k,u},\vec{x}_{i,u}\right)^2 - \mu_u = 0,$$

and thus

$$\begin{aligned} v_{i,k,u} &= \left(\frac{\mu_u}{m\, z_{i,u}^n\, d_u\left(\vec{w}_{k,u},\vec{x}_{i,u}\right)^2}\right)^{\frac{1}{m-1}} \quad (15) \\ \left(\frac{\mu_u}{m\, z_{i,u}^n}\right)^{\frac{1}{m-1}} &= v_{i,k,u}\left(d_u\left(\vec{w}_{k,u},\vec{x}_{i,u}\right)^2\right)^{\frac{1}{m-1}}. \quad (16) \end{aligned}$$

Zeroing the derivative of F_i w. r. t. μ_u will result in condition (4), ensuring that the partition values sum to 1, i. e.

$$\frac{\partial F_i}{\partial \mu_u} = 1 - \sum_{k=1}^{c_u} v_{i,k,u} = 0\,. \quad (17)$$

We use (15) and (17) to come up with

$$\begin{aligned} 1 &= \sum_{k=1}^{c_u}\left(\frac{\mu_u}{m\, z_{i,u}^n\, d_u\left(\vec{w}_{k,u},\vec{x}_{i,u}\right)^2}\right)^{\frac{1}{m-1}}, \\ 1 &= \left(\frac{\mu_u}{m\, z_{i,u}^n}\right)^{\frac{1}{m-1}}\sum_{k=1}^{c_u}\left(\frac{1}{d_u\left(\vec{w}_{k,u},\vec{x}_{i,u}\right)^2}\right)^{\frac{1}{m-1}} \quad (18) \end{aligned}$$

Equation (16) allows us to replace the first multiplier in (18). We will use the $\bar{k}$ notation to point out that the sum in (18) considers all partitions in a universe and k to denote one particular cluster coming from (15),

$$\begin{aligned} 1 &= v_{i,k,u}\left(d_u\left(\vec{w}_{k,u},\vec{x}_{i,u}\right)^2\right)^{\frac{1}{m-1}} \\ &\quad \times \sum_{\bar{k}=1}^{c_u}\left(\frac{1}{d_u\left(\vec{w}_{\bar{k},u},\vec{x}_{i,u}\right)^2}\right)^{\frac{1}{m-1}} \\ 1 &= v_{i,k,u}\sum_{\bar{k}=1}^{c_u}\left(\frac{d_u\left(\vec{w}_{k,u},\vec{x}_{i,u}\right)^2}{d_u\left(\vec{w}_{\bar{k},u},\vec{x}_{i,u}\right)^2}\right)^{\frac{1}{m-1}} \end{aligned}$$

Finally, the update rule for $v_{i,k,u}$ arises as (see also 6):

$$v_{i,k,u} = \frac{1}{\sum_{\bar{k}=1}^{c_u}\left(\frac{d_u(\vec{w}_{k,u},\vec{x}_{i,u})^2}{d_u(\vec{w}_{\bar{k},u},\vec{x}_{i,u})^2}\right)^{\frac{1}{m-1}}}.$$

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]:

$$d_u\left(\vec{w}_{k,u},\vec{x}_{i,u}\right)^2 = \sum_{a=1}^{A_u}\left(w_{k,u,a} - x_{i,u,a}\right)^2, \quad (19)$$

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

with A_u the number of dimensions in universe u and $w_{k,u,a}$ the value of the prototype in dimension a. $x_{i,u,a}$ is the value of the a-th 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 (19) we obtain

$$\begin{aligned}
\frac{\partial J_{m,n}}{\partial w_{k,u,a}} = 0 &= 2 \sum_{i=1}^{|T|} z_{i,u}^n \, v_{i,k,u}^m \, (w_{k,u,a} - x_{i,u,a}) \\
w_{k,u,a} \sum_{i=1}^{|T|} z_{i,u}^n \, v_{i,k,u}^m &= \sum_{i=1}^{|T|} z_{i,u}^n \, v_{i,k,u}^m \, x_{i,u,a} \\
w_{k,u,a} &= \frac{\sum_{i=1}^{|T|} z_{i,u}^n \, v_{i,k,u}^m \, x_{i,u,a}}{\sum_{i=1}^{|T|} z_{i,u}^n \, v_{i,k,u}^m} \,,
\end{aligned}$$

which is also given with (8).

References

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

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

[3] S. Bickel and T. Scheffer. Multi-view clustering. In *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 19–26, 2004.

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual Conference on Computational Learning Theory (COLT'98)*, pages 92–100. ACM Press, 1998.

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

[6] R. N. Dave. Characterization and detection of noise in clustering. *Pattern Recognition Letters*, 12:657–664, 1991.

[7] D. J. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. MIT Press, 2001.

[8] F. Höppner, F. Klawoon, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis*. John Wiley, Chichester, England, 1999.

[9] K. Kailing, H.-P. Kriegel, A. Pryakhin, and M. Schubert. Clustering multi-represented objects with noise. In *PAKDD*, pages 394–403, 2004.

[10] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newsl.*, 6(1):90–105, 2004.

[11] D. E. Patterson and M. R. Berthold. Clustering in parallel universes. In *Proceedings of the 2001 IEEE Conference in Systems, Man and Cybernetics*. IEEE Press, 2001.

[12] W. Pedrycz. Collaborative fuzzy clustering. *Pattern Recognition Letters*, 23(14):1675–1686, 2002.

[13] A. Schuffenhauer, V. J. Gillet, and P. 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):295–307, 2000.

[14] N. B. Venkateswarlu and P. S. V. S. K. Raju. Fast ISODATA clustering algorithms. *Pattern Recognition*, 25(3):335–342, 1992.

[15] J. Wang, H.-J. Zeng, Z. Chen, H. Lu, L. Tao, and W.-Y. Ma. ReCoM: Reinforcement clustering of multi-type interrelated data objects. In *In Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'03)*, pages 274–281, 2003.

[16] B. Wiswedel and M. R. Berthold. Fuzzy clustering in parallel universes. In *Proc. Conf. North American Fuzzy Information Processing Society (NAFIPS 2005)*, pages 567–572, 2005.

[17] R. R. Yager and D. P. Filev. Approximate clustering via the mountain method. *IEEE Trans. Systems Man Cybernet.*, 24(8):1279–1284, August 1994.