

Granular Prototyping in Fuzzy Clustering

Andrzej Bargiela, *Member, IEEE*, Witold Pedrycz, *Fellow, IEEE*, and Kaoru Hirota

Abstract—We introduce a logic-driven clustering in which prototypes are formed and evaluated in a sequential manner. The way of revealing a structure in data is realized by maximizing a certain performance index (objective function) that takes into consideration an overall level of matching (to be maximized) and a similarity level between the prototypes (the component to be minimized). The prototypes identified in the process come with the optimal weight vector that serves to indicate the significance of the individual features (coordinates) in the data grouping represented by the prototype. Since the topologies of these groupings are in general quite diverse the optimal weight vectors are reflecting the anisotropy of the feature space, i.e., they show some local ranking of features in the data space. Having found the prototypes we consider an inverse similarity problem and show how the relevance of the prototypes translates into their granularity.

Index Terms—Direct and inverse matching problem, granular prototypes, information granulation, logic-based clustering, similarity index, t - and s -norms.

I. INTRODUCTION

THERE is a wealth of clustering techniques [1], [3], [16], [24] and a diversity of ways in which clustering is used in fuzzy modeling and pattern recognition, cf. [12], [24], and [25]. Clusters are information granules and as such start playing a central role at the algorithmic layer of the technology of fuzzy sets. Granular computing is an important methodological endeavor and dwells quite substantially on fuzzy clustering, especially its niche addressing aspects of granular prototypes and granular constructs, in general. There have been several pursuits along this line [9], [19]–[21] yet the area is still in its early development stage. This study being in line of granular clustering proposes a comprehensive design toward logic-driven clustering culminating in a granular type of prototypes. There are several objectives we would like to formulate in this context. First, we would like to build prototypes in a sequential manner so that they could be ranked with respect to their relevance. Second, we would like the clustering algorithm to exhibit significant explorative capabilities. This will be instilled by defining a suit-

able performance index (objective function). Third, the way in which the prototypes are formed should lend itself to their granular extension.

In the paper, we proceed with a top-down presentation by first discussing the essence of the method and then elaborating on all pertinent details. The experimental part of the study consists of low-dimensional (mainly two-dimensional) patterns, as our intent is to illustrate the efficacies of the proposed clustering and granulation mechanisms. We contrast the algorithm with the fuzzy c -means (FCM) being treated as a *de-facto* standard in fuzzy clustering.

The material is organized into five sections. First, in Section II we formulate the problem and elaborate on the underlying terminology and notation (that is consistent with the one encountered in fuzzy sets). The two concepts fundamental to the general clustering approach are a notion of matching (comparison) of fuzzy sets and a construction of an objective function (performance index) guiding a way in which a structure in data is developed. Section III is devoted to prototype optimization where we show detailed derivations of explicit formulas for the prototypes. These derivations and resulting formulas imply a way in which the overall flow of computations goes: The essence of our approach can be summarized as iterative construction of clusters guided by the performance index (so that they can be added if appropriate) without any upfront commitment as to the number of clusters. This is in contrast to some other methods such as FCM. The development of a granular version of the prototypes, that builds on the numeric prototypes designed earlier is discussed in Section IV. It is shown that this design splits into two phases in which the performance index associated with each prototype is transformed into its granular (interval) envelope through solving an inverse matching problem. Conclusions are covered in Section V.f E

II. PROBLEM FORMULATION

The problem formulation comprises several main components such as a format of data, a form of the performance index and a general organization of the search for data.

In this study, we are concerned with data (patterns) distributed in an n -dimensional $[0,1]$ hypercube. In what follows, we will be treating the data as points in $[0,1]^n$, say $\mathbf{x} \in [0,1]^n$. In general we are concerned with N patterns (data points) $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$. The “standard” objective of the clustering method (no matter what is its realization) is to reveal a structure in the data set and to present it in a readable and easily comprehensible format. In general, we consider a collection of prototypes to be a tangible and compact reflection of the overall structure. In the approach undertaken here we adhere to the same principle. The prototypes representing each cluster are selected as some elements of the data set. Their selection is realized in such a way that

Manuscript received December 23, 2001; revised February 12, 2003 and December 9, 2003. This work was supported by the Engineering and Physical Sciences Research Council of the U.K. (EPSRC), by the Natural Sciences and Engineering Research Council of Canada (NSERC), and by the Alberta Software Engineering Research Consortium (ASERC).

A. Bargiela is with the Department of Computing and Mathematics, The Nottingham Trent University, Nottingham NG1 4BU, U.K. (e-mail: andre@doc.ntu.ac.uk)

W. Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2R3, Canada, and also with the Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland (e-mail: pedrycz@ee.ualberta.ca)

K. Hirota is with the Department of Computational Intelligence and Systems Science, Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama-city 226-8502, Japan.

Digital Object Identifier 10.1109/TFUZZ.2004.834808

they 1) match (represent) the data to the highest extent while 2) being evidently distinct from each other. These two requirements are represented in the objective function guiding the clustering process. In the sequel, we define the detailed components of the optimization. Since the elements in the unit hypercube can be viewed as fuzzy sets, we can take advantage of well-known logic operations developed in this domain. The notion of similarity (equality) between membership grades plays a pivotal role and this concept is crucial to the development of the clustering mechanisms.

A. Expressing Similarity Between Two Fuzzy Sets

The measure of similarity between two fuzzy sets (in this case a datum and a prototype) $\mathbf{x}=[x_1 \ x_2 \ \dots \ x_n]^T$ and $\mathbf{v}([v_1 \ v_2 \ \dots \ v_n]^T)$ is defined by incorporating the operation of matching (\equiv) encountered in fuzzy sets. The following definition will be used:

$$\text{sim}(\mathbf{x}, \mathbf{v}) = \prod_{i=1}^n (w_i^2 s(x_i \equiv v_i)). \quad (1)$$

In this, $T(\)$ and $s(\)$ denote a t-norm and s-norm, respectively. The weights (w_i) quantify an impact of each coordinate of the feature space $[0, 1]^n$ on the final value of the similarity index $\text{sim}(\)$. When convenient, we will be using a notation $\text{sim}(\mathbf{x}, \mathbf{v}; \mathbf{w})$ to emphasize the role played by the weight vector. The similarity between two membership grades is rooted in the fundamental concept of similarity (or equivalence) of two fuzzy sets (or sets). Given two membership grades a and b , (the values of a and b are confined to the unit interval), a similarity level $a \equiv b$ is computed in the form

$$a \equiv b = (a \rightarrow b)t(b \rightarrow a) \quad (2)$$

where the implication operation (\rightarrow) is defined as a residuation (ϕ -operator) [4], [5], [12], [13] that is

$$a \rightarrow b = \sup\{c \in [0, 1] | atc \leq b\}. \quad (3)$$

This expression of the residuation is induced by a certain t-norm. The implication models a property of inclusion; referring to (3) we note that it just quantifies a degree to which a is included in b . The *and* connective used in (2) translates it into a verbal expression

$$(a \text{ is included in } b) \text{ and } (b \text{ is included in } a) \quad (4)$$

which, in essence, quantifies an extent to which two membership grades are equal. As a matter of fact, the origin of this definition traces back to what we know well in set theory: We say two sets A and B are equal if A is included in B and B is included in A . Moving on with the definition, the visualization of the similarity treated as a function of “ a ” with “ b ” regarded as a parameter of this index is included in Fig. 1. As expected, it attains 1 if and only if a is equal to b . The function decreases when moving away from “ b .” However, quite asymmetric where

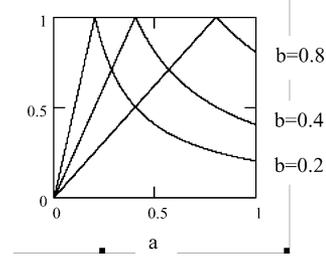


Fig. 1. Similarity index $a \equiv b$ regarded as a function of a for selected values of b . The residuation is induced by the product operation, $a \rightarrow b = \min(1, b/a)$.

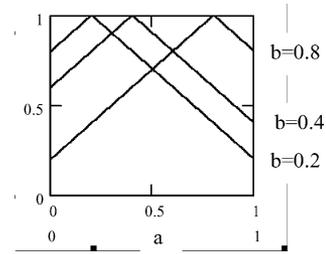


Fig. 2. Similarity index $a \equiv b$ regarded as a function of “ a ” for selected values of “ b ” and Lukasiewicz implication operation, $a \rightarrow b = \min(1, 1 - a + b)$.

this asymmetry arises as a consequence of the implication operations being used in the definition. Note also that the change in the t-norm in the basic definition (2) does not affect the form of the similarity index.

The similarity index is affected by the residuation operation (being more precise, a specific t-norm being used to induce it). For example, Lukasiewicz implication (induced by the Lukasiewicz t-norm) produces a series of piecewise linear plots, Fig. 2.

For some alternative definitions of similarity measures refer to [5].

The illustration of the similarity index in case of two variables ($n = 2$) is shown in Fig. 3. The intent is to visualize the impact of the weights on the performance of the index. It becomes apparent that high values of the weight reduce the impact of the corresponding variable.

B. Performance Index (Objective Function)

Performance index reflects the character of the underlying clustering philosophy. In this work we have adopted performance index that can be concisely described in the following manner. A prototype of the first cluster \mathbf{v}_1 is selected as one of the elements of the data set ($\mathbf{v}_1 = \mathbf{x}_j$ for some $j = 1, 2, \dots, N$) so that it maximizes the sum of the similarity measures of the form

$$\sum_{k=1}^N \text{sim}(\mathbf{x}_k, \mathbf{v}_1; \mathbf{w}_1) \Rightarrow \text{Max}_{\mathbf{v}_1, \mathbf{w}_1} \quad (5)$$

with $\text{sim}(\mathbf{x}_k, \mathbf{v}_1, \mathbf{w}_1)$ defined by (1). Once the first cluster (prototype) has been determined (through a direct search across the data space with a fixed weight vector and subsequent optimization of the weights treated as another part of the optimization

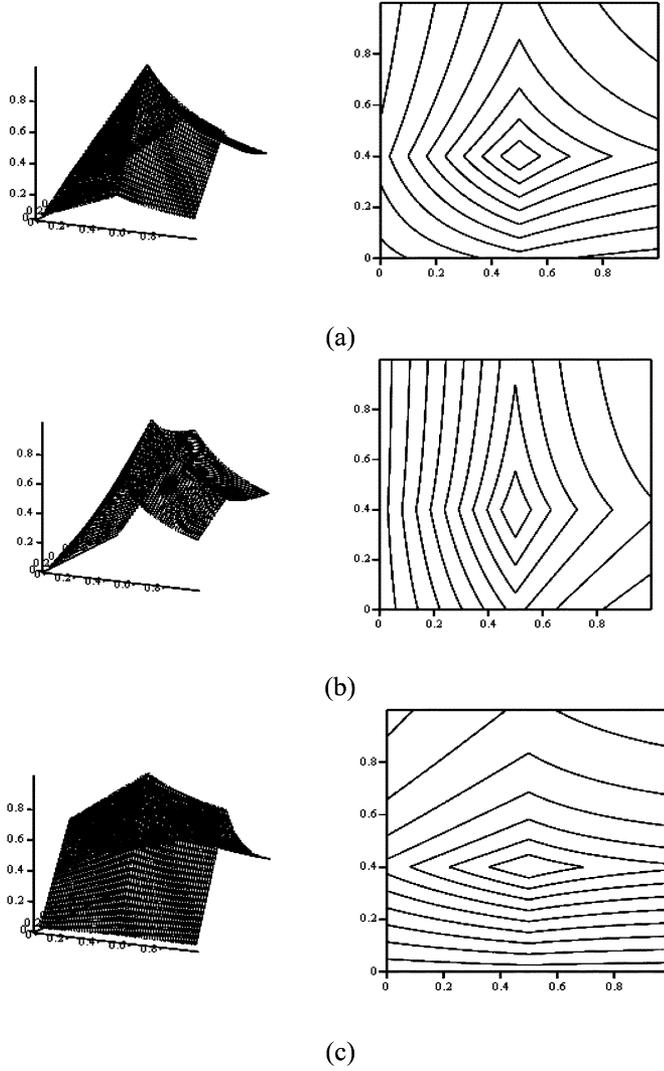


Fig. 3. Similarity index (three-dimensional plot and two-dimensional contours) for selected values of weight factors. (a) $w_1 = 0.5, w_2 = 0.5$. (b) $w_1 = 0.2, w_2 = 0.8$. (c) $w_1 = 0.8, w_2 = 0.2$. In all cases, $\mathbf{v} = [0.5, 0.4]$.

process), we move on to the next cluster (prototype) \mathbf{v}_2 and repeat the cycle. The form of the objective function remains the same throughout the iterative process but we combine now the maximization of the sum of similarity measures (5) with a constraint on the relative positioning of the new prototype. The point is that we want this new prototype, say \mathbf{v}_2 , not to “duplicate” the first prototype by being too close to it and thus not representing any new part of the data. To avoid this effect, we now consider the expression of the form

$$(1 - \text{sim}(\mathbf{v}_2, \mathbf{v}_1; \mathbf{0})) \sum_{k=1}^N \text{sim}(\mathbf{x}_k, \mathbf{v}_2; \mathbf{w}_2) \quad (6)$$

where the first factor $1 - \text{sim}(\mathbf{v}_1, \mathbf{v}_2; \mathbf{0})$ expresses the requirement of \mathbf{v}_2 to be as far apart from \mathbf{v}_1 as possible. The above expression has to be maximized with respect to \mathbf{v}_2 and this optimization has to be carried out with the weight vector (\mathbf{w}_2) involved. In the sequel, we proceed with the determination of the

third prototypes \mathbf{v}_3 , etc. In general, the optimization of the L -th prototype follows the expression:

$$Q(L) = (1 - \text{sim}(\mathbf{v}_L, \mathbf{v}_{L-1}; \mathbf{0}))(1 - \text{sim}(\mathbf{v}_L, \mathbf{v}_{L-2}; \mathbf{0})) \cdots (1 - \text{sim}(\mathbf{v}_L, \mathbf{v}_1; \mathbf{0})) \sum_{k=1}^N \text{sim}(\mathbf{x}_k, \mathbf{v}_L; \mathbf{w}_L). \quad (7)$$

As noted, this expression takes into account all previous prototypes when looking for the current prototype. Interestingly, the performance index to be maximized is a decreasing function of the prototype index, that is $L_1 < L_2$ implies that $Q(L_1) \leq Q(L_2)$.

Another observation of interest is that the first prototype constitutes the best representative of the overall data set. Subsequent prototypes are, in effect, the best representatives for the more detailed partitions of data.

So far, we have not touched the issue of the optimization of the weight vector associated with the prototype that is an integral part of the overall clustering. The next section provides a solution to this problem.

III. PROTOTYPE OPTIMIZATION

Let us concentrate on the optimization of the performance index in its general form given by (7). Apparently the optimization consists of two phases, that is: 1) the determination of the prototype (\mathbf{v}_L), and 2) the optimization of the weight vector (\mathbf{w}_L). These two phases are intertwined yet they exhibit a different character. The prototype is about enumeration out of a finite number of options (patterns in the data set). The weight optimization has not been formulated in detail and now requires a prudent formulation as a constraint type of optimization (without any constraint the task may return a trivial solution).

Referring to (7) we observe that it can be written down in the form

$$Q(L) = G \sum_{k=1}^N \text{sim}(\mathbf{x}_k, \mathbf{v}_L; \mathbf{w}_L). \quad (8)$$

Note that the first part of the original expression does not depend on \mathbf{w}_L and can be treated as constant with this regard

$$G = (1 - \text{sim}(\mathbf{v}_L, \mathbf{v}_{L-1}; \mathbf{0}))(1 - \text{sim}(\mathbf{v}_L, \mathbf{v}_{L-2}; \mathbf{0})) \cdots (1 - \text{sim}(\mathbf{v}_L, \mathbf{v}_1; \mathbf{0})). \quad (9)$$

We impose the following constraint on \mathbf{w}_L requesting that its components are located in the unit interval and sum up to 1:

$$\sum_{j=1}^n w_{Lj} = 1. \quad (10)$$

The optimization of (8) with respect to \mathbf{w}_L for a fixed prototype \mathbf{v}_L is expressed as

$$\begin{aligned} \max Q(L) &= G \sum_{k=1}^N \text{sim}(\mathbf{x}_k, \mathbf{v}_L; \mathbf{w}_L) \\ \text{subject to} \quad &\sum_{j=1}^n w_{Lj} = 1. \end{aligned} \quad (11)$$

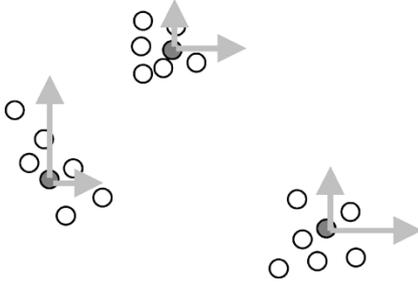


Fig. 4. Anisotropy of the feature space of patterns represented by weight vectors associated with prototypes.

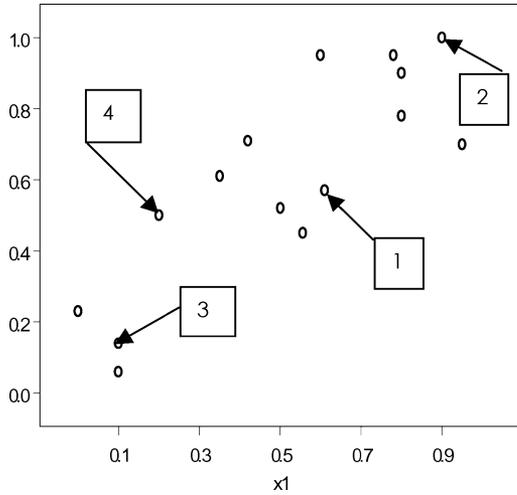


Fig. 5. Synthetic data; successive detected prototypes are identified by arrows and corresponding numbers.

The detailed derivations of the weight vector is done through the technique of Lagrange multipliers. First, we form an augmented form of the performance index

$$V = G \sum_{k=1}^N \left\{ \prod_{j=1}^n (w_{L_j}^2 s(x_{kj} \equiv v_{L_j})) \right\} - \lambda \left(\sum_{j=1}^n w_{L_j} - 1 \right). \quad (12)$$

To shorten the expression, we introduce the notation $u_{ks} = x_{ks} \equiv v_{L_s}$. The derivative of V taken with respect to w_{L_s} (the s th coordinate of the weight vector) is set to zero and the solution of the resulting equations gives rise to the optimal weight vector

$$\frac{dV}{dw_{L_s}} = 0 \quad \frac{dV}{d\lambda} = 0. \quad (13)$$

The derivatives can be computed once we specify t - and s -norms. For the sake of further derivations (and ensuing experiments), we consider a product and probabilistic sum as the corresponding models of these operations. Furthermore we introduce the abbreviated notation $u_{ks} = x_{ks} \equiv v_{L_s}$ for $k = 1, 2, \dots, N$ and $s = 1, 2, \dots, n$. Taking all of these into account, we have

$$\frac{dV}{dw_{L_s}} = G \sum_{k=1}^N \frac{d}{dw_{L_s}} \{ A_{ks} w_{L_s}^2 s u_{ks} \} - \lambda = 0 \quad (14)$$

TABLE I

PROTOTYPES AND THEIR CHARACTERIZATION; THE STARTING POINT WHERE THE VALUES OF THE PERFORMANCE INDEX STABILIZE HAS BEEN HIGHLIGHTED

Cluster no.	Prototype	Performance index	\mathbf{w}
1	[0.61 0.57]	8.490796	[0.43 0.57]
2	[0.90 1.00]	4.755118	[0.37 0.63]
3	[0.10 0.14]	3.523316	[0.25 0.75]
4	[0.20 0.50]	2.951507	[0.20 0.80]
5	[0.00 0.00]	2.193254	[0.30 0.70]
6	[0.00 0.23]	2.024915	[0.11 0.89]
7	[0.10 0.06]	1.845740	[0.26 0.74]

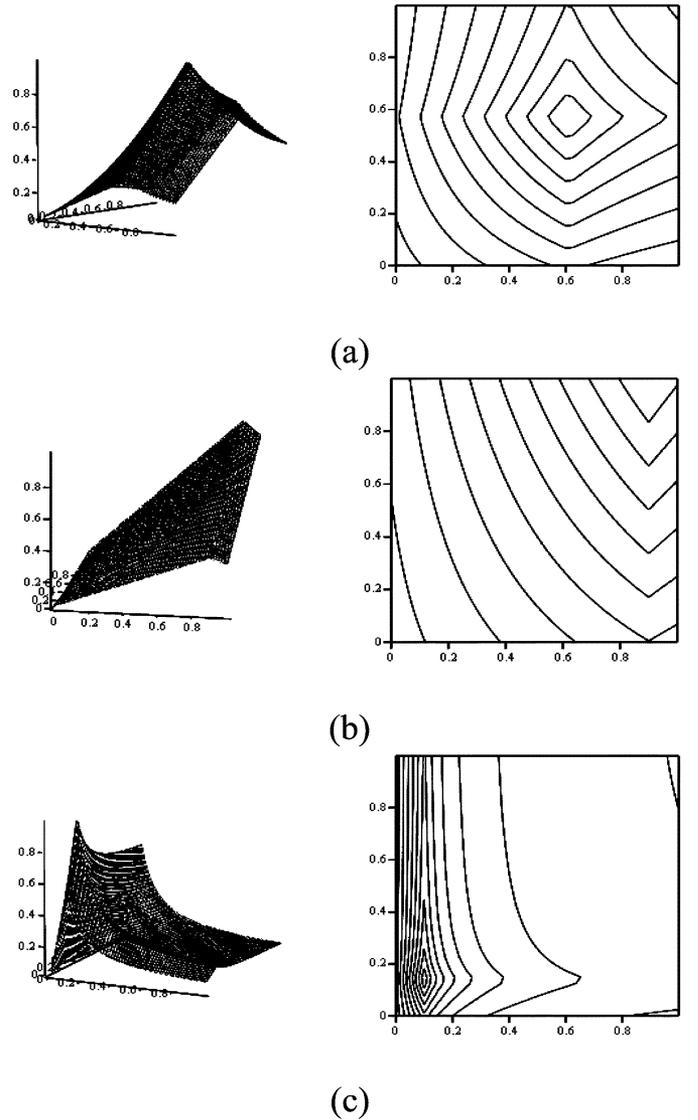


Fig. 6. Visualization (three-dimensional and contour plots) of the first three clusters in the feature space: cluster no. 1 (a), no. 2 (b), and 3 (c).

where

$$A_{ks} = \prod_{\substack{j=1 \\ j \neq s}}^n (w_{L_j}^2 s u_{ks}).$$

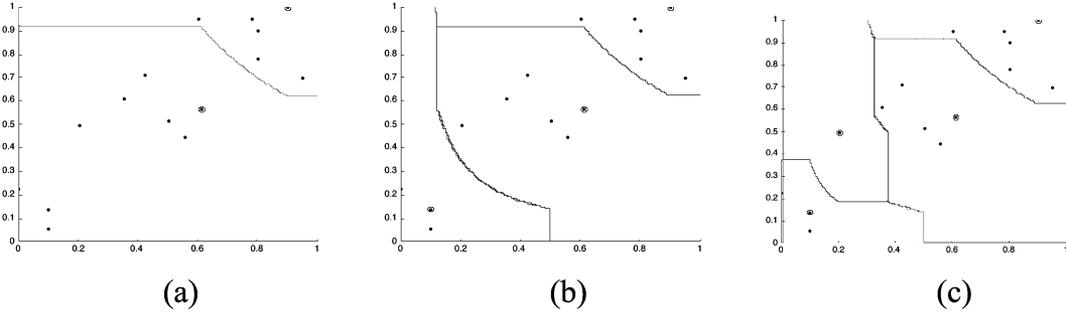


Fig. 7. Classification regions for (a) 2 clusters, (b) 3 clusters, and (c) 4 clusters, identified through maximization of the similarity measure.

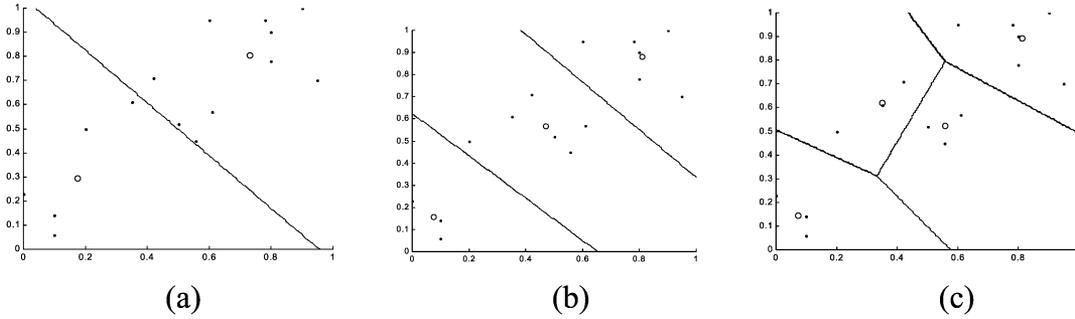


Fig. 8. FCM clustering and the implied partition of the pattern space for (a) $c = 2$, (b) $c = 3$, and (c) $c = 4$, clusters.

The use of the probabilistic sum (s-norm) in (14) leads to the expression

$$\begin{aligned} \frac{d}{dw_{Ls}} \{A_{ks} w_{Ls}^2 s u_{ks}\} &= A_{ks} \frac{d}{dw_{Ls}} (w_{Ls}^2 + u_{sk} - w_{Ls}^2 u_{sk}) \\ &= 2A_{ks} w_{Ls} (1 - u_{ks}) \end{aligned} \quad (15)$$

and, in the sequel

$$\frac{dV}{dw_{Ls}} = 2G w_{Ls} \sum_{k=1}^N A_{ks} (1 - u_{ks}) - \lambda = 0. \quad (16)$$

From (16), we have

$$w_{Ls} = \frac{\lambda}{2G \sum_{k=1}^N A_{ks} (1 - u_{ks})}. \quad (17)$$

The form of the constraint, $\sum_{j=1}^c w_{Lj} = 1$, produces the following expression:

$$\frac{\lambda}{2} \sum_{j=1}^c \frac{1}{G \sum_{k=1}^N A_{kj} (1 - u_{kj})} = 1 \quad (18)$$

or

$$\frac{\lambda}{2} = \frac{1}{\sum_{j=1}^c \frac{1}{G \sum_{k=1}^N A_{kj} (1 - u_{kj})}}. \quad (19)$$

Finally, inserting (19) into (17), the s th coordinate of the weight vector reads as

$$w_{Ls} = \frac{1}{\sum_{j=1}^c \frac{\sum_{k=1}^N A_{ks} (1 - u_{ks})}{\sum_{k=1}^N A_{kj} (1 - u_{kj})}}. \quad (20)$$

Summarizing the algorithm, it essentially consists of two steps. We try every pattern as a potential prototype, for each choice we optimize the weights and find a maximal value of $Q(L)$ out of N options available. The one that maximizes this performance index is treated as a prototype. It comes with an optimal weight vector w_L . Each prototype comes with its own weight vector that may vary from prototype to prototype. Bearing in mind the interpretation of these vectors we can say that they articulate the “local” characteristics of the feature space of the patterns. As seen in Fig. 3, the lower the value of the weight for a certain feature (variable), the more essential the corresponding feature is. Significantly, the importance of the features is not the same across the entire space. The space becomes highly anisotropic where prototypes come equipped with different ranking of the features, see Fig. 4. This is in fact quite intuitive since the topology of the local data groupings represented by the prototypes implies that some features (dimensions) are more representative in this local context than others.

The computational complexity of the aforementioned algorithm is of the order $O(N^2 n^2)$, since each pattern is considered a potential prototype and there are $N - 1$ evaluations of the similarity measure for each prototype. Also the evaluation of each of the n coordinates of the optimum weights (20) involves summation of n factors thus giving rise to the n^2 factor in the complexity order. Consequently, the application of the algorithm to

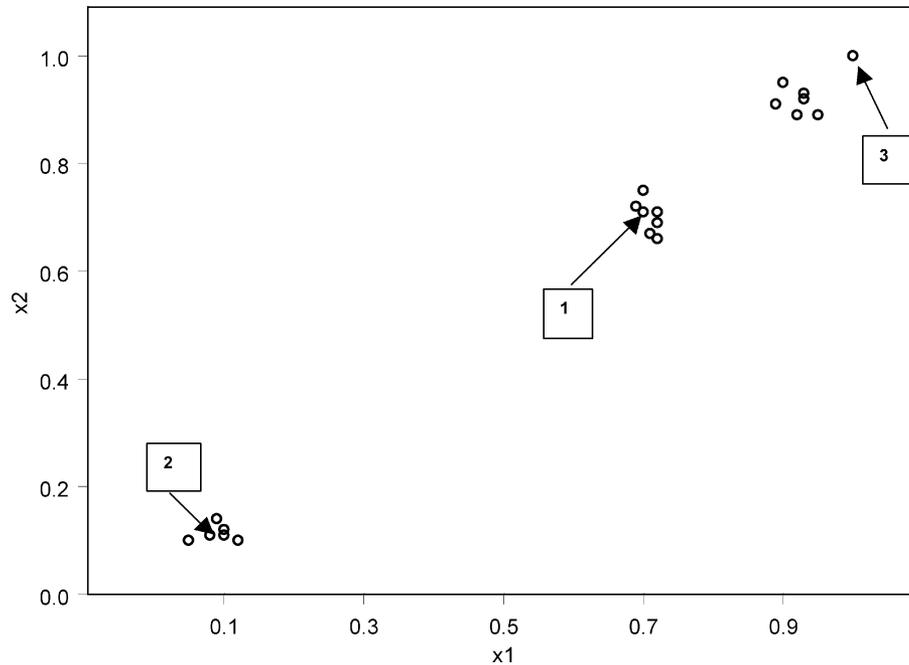


Fig. 9. Two-dimensional synthetic data with three first prototypes identified by the clustering algorithm.

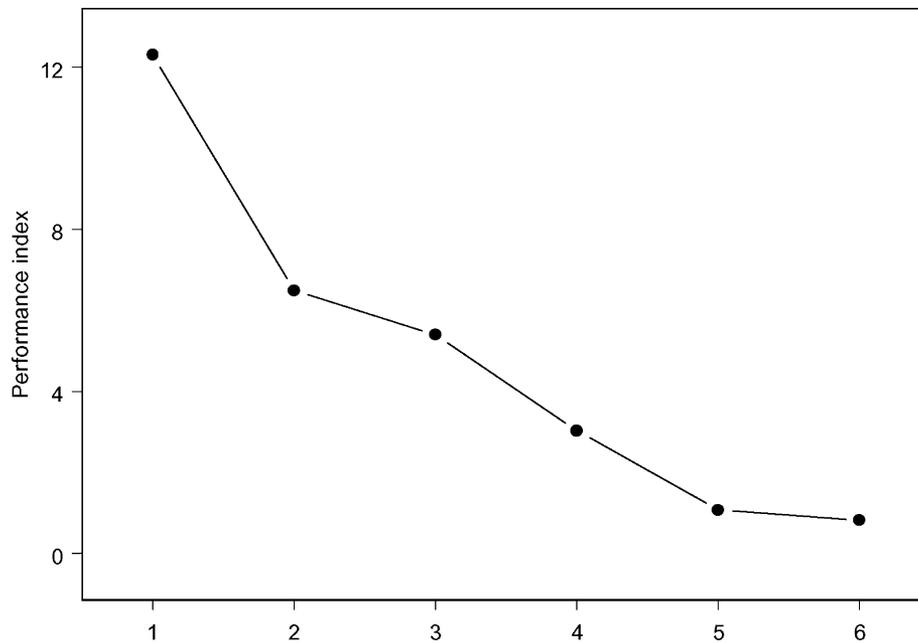


Fig. 10. Performance index versus number of clusters (c).

large data set is not practical and would necessitate either data partitioning or deployment of some heuristics that would reduce the number of evaluations of the similarity measure.

In what follows, we discuss a number of low-dimensional synthetic data sets that help us grasp the meaning of the resulting prototypes and interpret their weights.

Example 1: The two-dimensional data set shown in Fig. 5 exhibits several not very strongly delineated clusters.

The clustering is completed out by forming an additional cluster once at a time. The values of the performance index associated with the clusters, a position of the prototypes and their respective weights are summarized in Table I. As expected, the performance of successive clusters gets lower and the proto-

types start locating themselves close to each other. This feature of the clustering approach helps us investigate the relevance of the clusters on the fly and stop the search for more structure once the respective performance indexes start assuming low values. In this example this happens for $c = 5$ at which point the values of the performance index stabilize.

Table I includes also the weight vectors associated with the prototypes. They reflect upon the “local” properties of the feature space. From their analysis (recall that lower value of the weight means higher relevance of the feature in the neighborhood of the given prototype), we learn that the first feature (x_1) is more relevant than the second one. This is a quantification of the visual inspection: As seen from Fig. 5, when projecting the

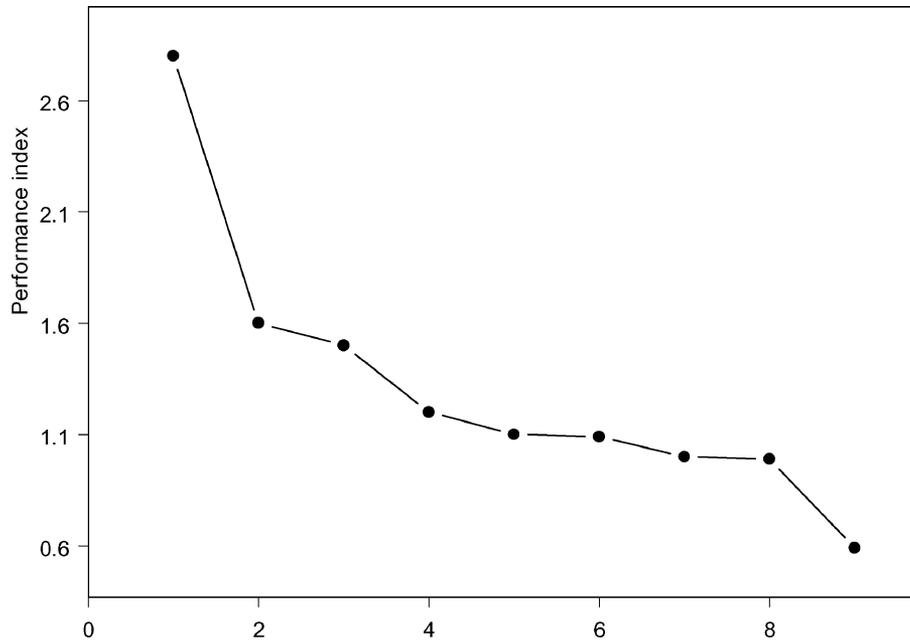


Fig. 11. Performance index versus number of clusters (c).

TABLE II
FOUR-DIMENSIONAL SYNTHETIC PATTERNS

Pattern no.	Coordinates	Prototype
1	0.80 0.10 0.60 0.30	
2	0.50 0.20 0.40 0.31	
3	0.60 0.30 0.10 0.35	
4	0.40 0.18 0.87 0.40	←2
5	0.90 0.15 0.50 0.32	←4
6	0.20 0.95 0.65 0.30	
7	0.20 0.40 0.30 0.31	←3
8	0.70 0.20 0.63 0.28	←1
9	1.00 0.00 1.00 0.31	
10	0.05 0.15 0.42 0.33	

data on x_2 they tend to be more “crowded” (start overlapping) in comparison with their projection on x_1 .

The prototypes produce nonlinear classification boundaries as shown in Fig. 7.

For comparative reasons we carried out clustering using FCM; the resulting prototypes and the boundaries between the clusters are included in Fig. 8. It can be seen that the nonlinear boundaries between the clusters identified through maximization of the similarity measure afford much more refined partition of the pattern space.

Example 2: This two-dimensional data, Fig. 9 shows a structure that has three condensed clusters but also includes two points that are somewhat apart from the clusters.

The results are included in Fig. 9. The values of the performance index are visualized in Fig. 10. It can be seen that the performance index “flattens-out” for five clusters, which corresponds to identifying significantly distinct data groupings.

Example 3: The four dimensional data are given in Table II.

TABLE III
WEIGHT VECTORS OF THE FIRST FOUR PROTOTYPES

prototype no.	weight vector
1	0.12 0.12 0.16 0.60
2	0.16 0.18 0.14 0.52
3	0.04 0.04 0.06 0.86
4	0.06 0.09 0.15 0.70

The “optimal” number of clusters is equal to 4 (at this number we see “flattening-out” of the values of the performance index, which means that the maximization of the similarity between data and prototypes is counterbalanced by the increase of similarity between the prototypes), Fig. 11. The weight vectors of the prototypes, Table III, tell an interesting story: The feature space is quite isotropic and in all cases the first feature (x_1) carries a higher level of relevance (the first coordinate of weight vector of each prototype is constantly lower than the other). This is highly intuitive as the patterns are more “distributed” along the first axis (x_1), which makes it more relevant (discriminatory) in this problem.

Example 4: This two-dimensional data set reveals two very unbalanced clusters—the first group is evidently dominant (100 patterns) over the second cluster (which consists of five data points), Fig. 12.

As we start building the prototypes, they start representing both clusters in more detail. The second prototype in the sequence has been assigned to the small cluster meaning that the method is after some still not represented parts of the data structure. We may say that the form of the performance index promotes a vigorous exploration of the data space and acts against “crowding” of the clusters in a close vicinity of each other. The consecutive clusters are after the details of the larger cluster as they start unveiling some substructures. Noticeably, the sixth prototype is assigned to the small cluster, Table IV.

It is instructive to compare these results with the structure revealed by the FCM. As anticipated (and this point was raised in

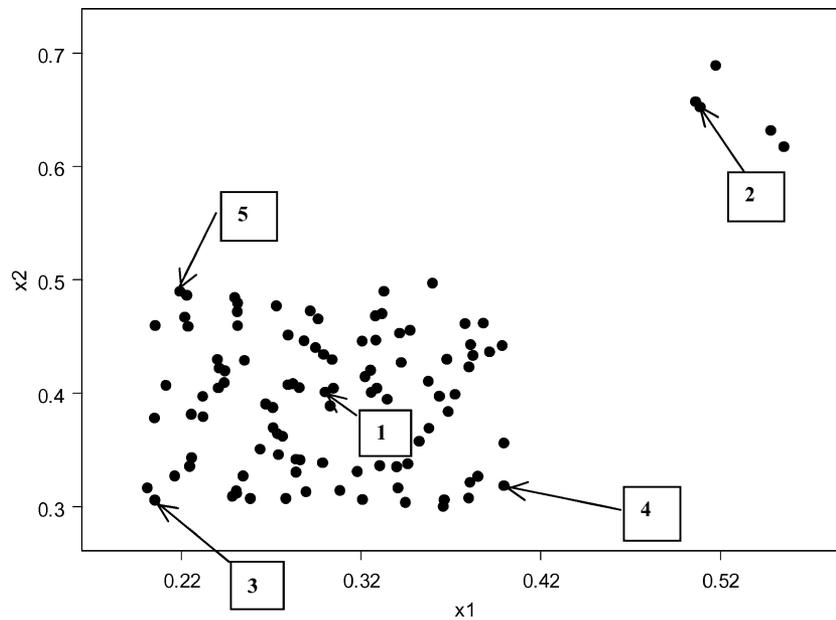


Fig. 12. Two-dimensional data set with two unequal clusters; the consecutive prototypes produced by the method were identified by numbers.

TABLE IV

PROTOTYPES OF THE CLUSTERS, THEIR PERFORMANCE INDEX AND WEIGHT VECTORS. THE SHADOWED ROW HIGHLIGHTS A SHARP DROP IN THE VALUES OF THE PERFORMANCE INDEX

Prototype no.	Location	Performance index	Weight vector
1	(0.300100 0.400800)	83.694626	[0.44 0.56]
2	(0.508700 0.652100)	33.989677	[0.48 0.52]
3	(0.205200 0.305700)	26.135773	[0.39 0.61]
4	(0.399500 0.318400)	9.132071	[0.42 0.58]
5	(0.219200 0.489600)	5.200340	[0.42 0.58]
6	(0.555300 0.617200)	1.585717	[0.41 0.59]
7	(0.359900 0.496900)	0.598020	[0.51 0.48]

the literature), FCM ignores the smaller cluster and it becomes primarily focused on the larger cluster. Only with the increase of the number of the clusters we start capturing the smaller of the clusters yet it happens later than we have reported in the previous method.

Example 5: The glass data set comes from the repository of Machine Learning¹, and concerns classification of several categories of glass. The study was motivated by criminological investigations. There are nine attributes (features) that are used in the classification, e.g., refractive index and a content of iron, magnesium, aluminum, etc. in the samples. There are seven classes (categories) identified in the problem.

In the experiment, we use first 100 patterns. The performance index for the individual prototypes is shown in Fig. 14. The plausible number of clusters is 5 since the performance index again “flattens-out” for larger number of clusters.

As the weight vectors of the individual prototypes are concerned (we confine ourselves to five most dominant prototypes) they show some level of anisotropy with the features being ranked quite consistently in the context of the individual prototypes. The mean values and standard deviations of the weights of the first five prototypes are shown as follows:

- mean values

0.059 757 0.025 592 0.055 397 0.063 844 0.053 884 0.070 637
0.042 023 0.610 937 0.018 282

- standard deviations

0.027 003 0.013 059 0.024 531 0.030 855 0.020 684 0.039 291
0.024 701 0.162 367 0.007 705.

Feature no. 8 can be clearly identified as relatively insignificant while the most essential ones are {2, 7, 9}. Their standard deviation is also quite low.

Example 6: A synthetic two-dimensional data set considered in this example is similar to the one analyzed in Example 4 but it has been tuned to illustrate the operation of granular prototyping on elongated data groupings. There are two data groupings. The first one consists of 100 patterns randomly scattered in a box [0.15 0.25; 0.50 0.90] and the second one consists of five patterns randomly scattered in a box [0.5 0.6; 0.2 0.24]. It is clear that with these topologies of individual data groupings the first feature is more discriminatory than the second one. The granular prototyping returned two prototypes $v_1 = [0.1553 \ 0.6774]$ and $v_2 = [0.5522 \ 0.6019]$ with the corresponding weights $w_1 = [0.3310 \ 0.6690]$ and $w_2 = [0.2040 \ 0.7960]$; see Fig. 15(a). Therefore, the numerical results give support to the intuition about the discriminatory value of the first feature (coordinate).

The property of the invariance of the granular prototyping to the rotation of the axis is illustrated in the second experiment with this data. The first and second coordinates are swapped round for all the patterns and the granular prototyping is rerun. The results are illustrated in Fig. 15(b). The granular prototypes identified for the modified data are $v_1 = [0.6774 \ 0.1553]$ and $v_2 = [0.6019 \ 0.5522]$ with the corresponding weights $w_1 =$

¹<http://www.ics.uci.edu/~mllearn/MLRepository.html>

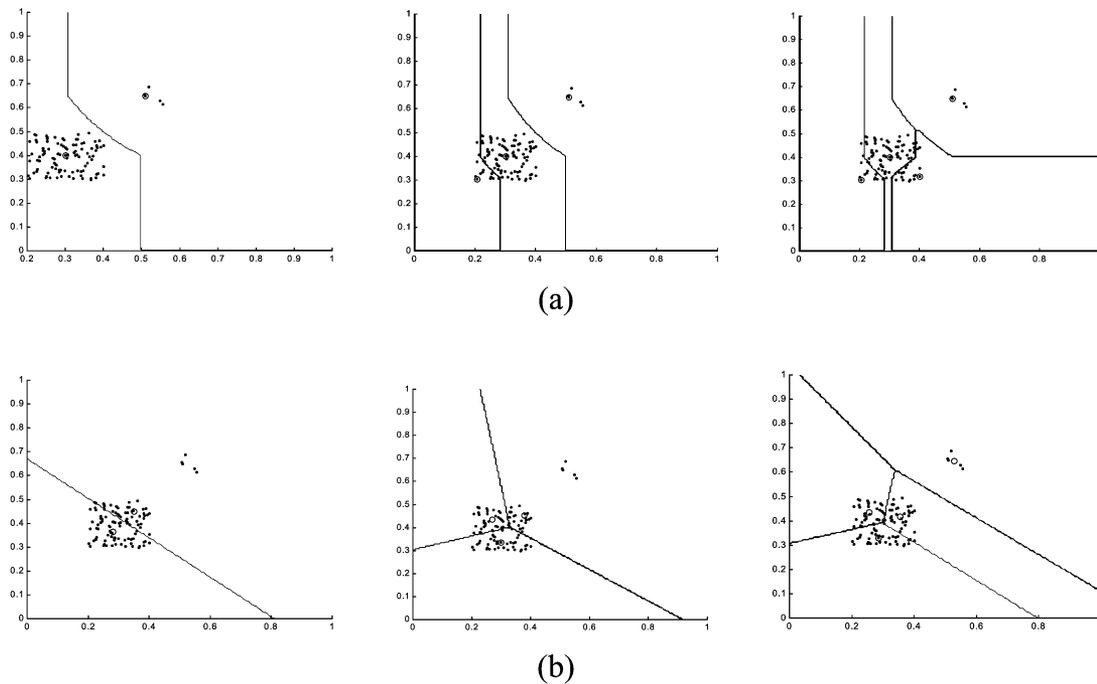


Fig. 13. Partition of the pattern space implied by (a) the similarity measure based clustering, and (b) the FCM clustering; for 2–4 clusters.

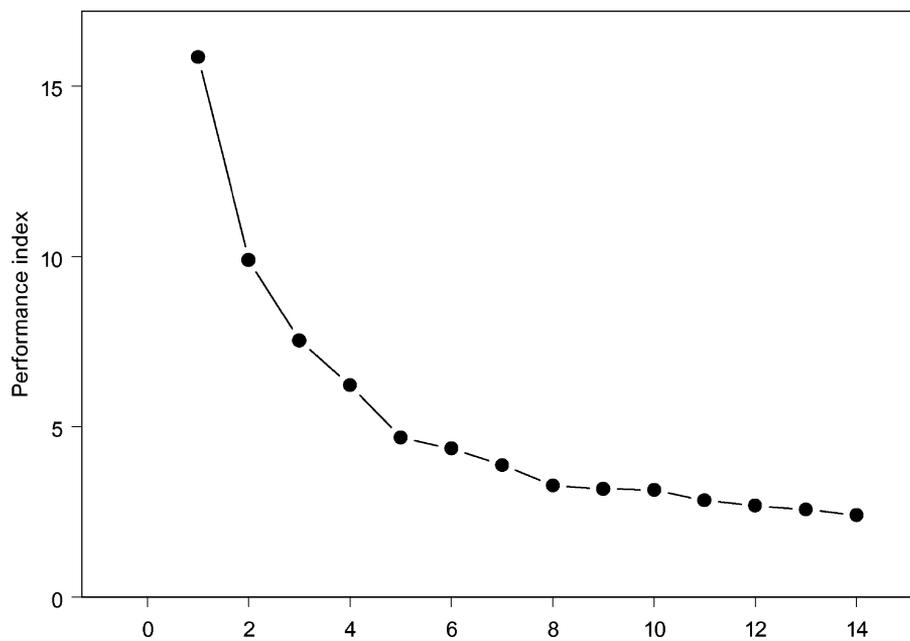


Fig. 14. Performance index versus number of clusters.

$[0.6690 \ 0.3310]$ and $w_2 = [0.7960 \ 0.2040]$. It is easy to note that these are the symmetrical reflections of the prototypes identified earlier. The same comment applies to the optimal weights associated with the prototypes.

IV. DEVELOPMENT OF GRANULAR PROTOTYPES

The inherently logic nature of the clustering technique helps us handle another interesting issue arising in the context of data summarization (and the clustering per se is aimed at this important target). As has become obvious from the previous sections,

the prototypes like the original data, are elements in the unit hypercube. One may question whether this is the only valid way of their representation. Naturally, one could expect that the prototypes as a form of summarization of the data should reflect the fact that the patterns being represented by them occupy a certain region in the feature space. This naturally lends itself to the notion of a granular prototype, that is a prototype that spreads in the feature space where its spread is related to the spatial characteristics of the original data. In a nutshell, we would like to develop prototypes that are represented as Cartesian products of intervals in the feature space. Our anticipation is that the gran-

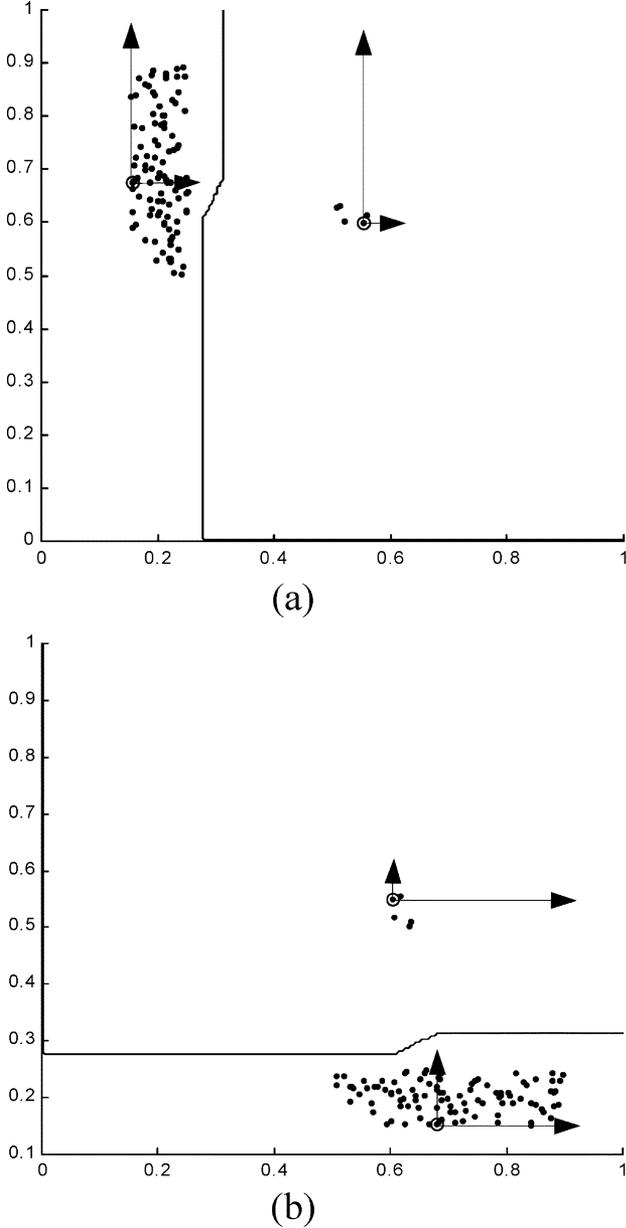


Fig. 15. Granular prototypes for the elongated data groupings. (a) Original set and (b) the set with rotated coordinates (b). Note that the coordinates of the weight vector have been scaled and preserve only their relative magnitude.

ularity of the prototypes gives us a better insight into the nature of the data as well as the relevance of the prototype itself. The formal framework of building granular prototypes can be introduced as follows. Consider \mathbf{v} to be the prototype $\mathbf{v} \in [0, 1]^n$ already determined in the way discussed in Section III. It comes with its weight vector \mathbf{w} . We can compute an average of similarity (q) between this prototype and all patterns by taking the following sum:

$$q = \frac{1}{N} \sum_{k=1}^N \text{sim}(\mathbf{x}_k, \mathbf{v}; \mathbf{w}) \quad (21)$$

(note that (21) is analogous to (8) with an exception that we do not consider here an interaction of \mathbf{v} with other prototypes and

that we normalize the result). This average similarity serves as a useful indicator of the relevance of the prototype. Now, let us determine such values of u_i for which (21) holds. As u_i is effectively a similarity level between x_i and v_i , in essence it implies the interval built around v_i . To see this, note that $u_i = x_i \equiv v_i$ so if v_i and u_i are given, one can determine the range into which x_i should fall in order to satisfy this equality. This range is just an interval (along i -th coordinate) that contains the prototype.

To form the granular prototype the process is repeated for all features, $i = 1, 2, \dots, n$, and we formulate and handle explicitly two optimization tasks arising here. The first one concerns the determination of the values of u_i , $i = 1, \dots, n$, so that they satisfy (21). The second task is an inverse problem emerging in the setting of the similarity index.

A. Optimization of the Similarity Levels

As a part of the construction of the granular prototypes, we encounter the problem of determining the matching levels along individual features given the weight vector \mathbf{w} and the overall matching level q . In other words, we are looking for $\mathbf{u} = [u_1 \ u_2 \ \dots \ u_n]$ such that

$$\frac{n}{T} \sum_{j=1}^n (w_j^2 s u_j) = \gamma \quad (22)$$

where \mathbf{u} collects the matching levels between given prototype \mathbf{v} and some other pattern \mathbf{x} . The aforementioned problem is not trivial and no closed form solution can be derived. Some iterative optimization should be deployed here. Bearing this in mind, we reformulate (22) as a standard MSE approximation problem

$$P = \left[\frac{n}{T} \sum_{j=1}^n (w_j^2 s u_j) - \gamma \right]^2 \rightarrow \text{Min}(\mathbf{u}) \quad (23)$$

whose solution is obtained by a series of modifications of \mathbf{u} through the gradient-based scheme, namely

$$\mathbf{u}(\text{new}) = \mathbf{u} - \alpha \nabla_{\mathbf{u}} P \quad (24)$$

where α denotes a positive learning rate. The detailed expression for the update can be derived for some predefined form of the triangular norm. Again using the product and probabilistic sum we produce a detailed expression for the gradient

$$u_k(\text{new}) = u_k - \alpha \frac{\partial P}{\partial u_k} \quad (25)$$

$k = 1, 2, \dots, n$. The detailed expression for the derivative is given as

$$\frac{\partial P}{\partial u_k} = 2 \left[\frac{n}{T} \sum_{j=1}^n (w_j^2 s u_j) - \gamma \right] \frac{\partial}{\partial u_k} \left(\frac{n}{T} \sum_{j=1}^n (w_j^2 s u_j) \right).$$

The inner derivative can be handled for specific t - and s -norm. For a certain pair of them (t -norm: product, s -norm: probabilistic sum), we have

$$\frac{\partial}{\partial u_k} \left(\frac{n}{T} \sum_{j=1}^n (w_j^2 s u_j) \right) = \frac{\partial}{\partial u_k} (B_k(w_k^2 + u_k - w_k^2 u_k)) = B_k(1 - w_k^2)$$

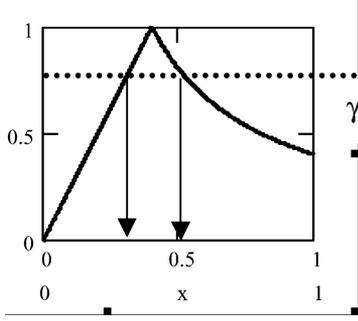


Fig. 16. Inverse matching problem: computing an interval of solutions to $x \equiv b \leq \gamma$.

where B_k computes using t-norm when excluding the index of interest (k)

$$B_k = \prod_{\substack{j=1 \\ j \neq k}}^n (w_j^2 s u_j).$$

B. Inverse Similarity Problem

The inverse problem coming with the similarity index can be formulated as follows: given b and γ (both in the unit interval), determine all possible values of “ x ” such that $x \equiv b = \gamma$. The character of the solution can be easily envisioned by augmenting this equality by its graphical interpretation, Fig. 16.

This figure underlines that the problem being formulated as before requires some refinement in order to enhance the interpretability of the solution and assure that it always exist. This can be done by moving from the equality to the inequality format of the relationship

$$x \equiv b \leq \gamma. \quad (26)$$

The solution to it arises in a form of a confidence interval (or simply interval) implied by a certain value of γ . This solution (interval) is a manifestation of the granularity of the prototype for a given feature. The solution to (26) can be obtained analytically for a specific type of the t-norm (or implication). As shown in Fig. 16, the solution always exists (that is there is always a nonempty interval for any given value of γ). The granularity of the prototype is a monotonic function of γ : higher values of γ imply higher values of granularity, i.e., narrow intervals of the granular prototype. For some critical (low enough) value of γ , the interval expands to the entire unit interval so we have a granular prototype of the lowest possible level of granularity.

Moving on to the detailed calculations, the interval of the granular prototype $[x_-, x_+]$ is equal to

$$\text{for } x \rightarrow b = \min\left(1, \frac{b}{x}\right)$$

$$x_- = \gamma b \quad x_+ = \min\left(1, \frac{b}{\gamma}\right) \quad (27)$$

$$\text{for } x \rightarrow b = \min(1, 1 - x + b)$$

$$x_- = \max(0, \gamma - 1 + b), \quad x_+ = \min(1, 1 - \gamma + b) \quad (28)$$

(these expressions are determined by considering the increasing and decreasing portions of the matching index as illustrated in Fig. 15).

Continuing the previous examples the resulting granular prototypes are shown in Figs. 17 and 18 for Examples 1 and 2, respectively. The same granular prototypes summarized as triples of the form {lower_bound, mode, upper_bound} are included in Table V. Note that by the mode we mean an original numeric value around which the granular prototype is constructed. The optimization of the degrees matching (\mathbf{u}) was completed by running the gradient based learning with $\alpha = 0.05$ for 100 iterations. The initial values of \mathbf{u} 's are set up as small (near zero) random numbers.

These granular prototypes reinforce and quantify our perception of structural dependencies in data. In the first case, Fig. 17, we note that the first component of the structure resides in the right upper quadrant of the coordinates and this shows very clearly in the distribution of the granules. As a matter of fact prototype 1 and 2 overlap (meaning that there is some redundancy). The next granule (implied by the third cluster) is essential to the quantification of the structure; it occupies the area close to the origin. The fourth cluster overlaps the third one. Noticeably, all granules are elongated along the second variable and this very much quantifies our observation about the limited relevance of this variable (note that all corresponding weights for the second variable are substantially high). The conclusion is that the granules tend to “expand” and occupy the space wherever it is possible; this expansion is visible for x_2 . The granular character of the prototypes in Fig. 18 is again a meaningful manifestation of the structure. The first two granules are far apart (and represent the two evidently distinct groups of data). The boxes do not discriminate between the variables viewing them as equally essential. The third granule overlaps with the first one as these two clusters are relatively close. The fourth cluster has a strong resemblance (and overlap) to the second granule.

As this analysis reveals, we can envision a structure of the data by inspecting the resulting granular prototypes. First, these granules help us position clusters in the data space (it is worth stressing that the numeric representation does not support this form of analysis). Second, we can envision a general geometry of data that could be helpful in the design of more detailed classifiers or other models.

The granules may exhibit some level of overlap (no matter how such overlap is expressed in a formal fashion). This may help reason about possible relevance and redundancy of some of these clusters.

V. CONCLUSION

We have introduced a new logic-based approach to data analysis by building a certain clustering environment. Their main and unique features worth underlining include the following.

- Logic-based character of processing: The search for structure in data is accomplished by exploiting fuzzy set operations. In particular, this concerns the matching operation that is easily interpretable and comes with a well defined semantics.
- Successive (sequential) construction of the prototypes and an assessment of their representation capabilities: The number of the clusters is not fixed in advance but can be adjusted dynamically depending upon the

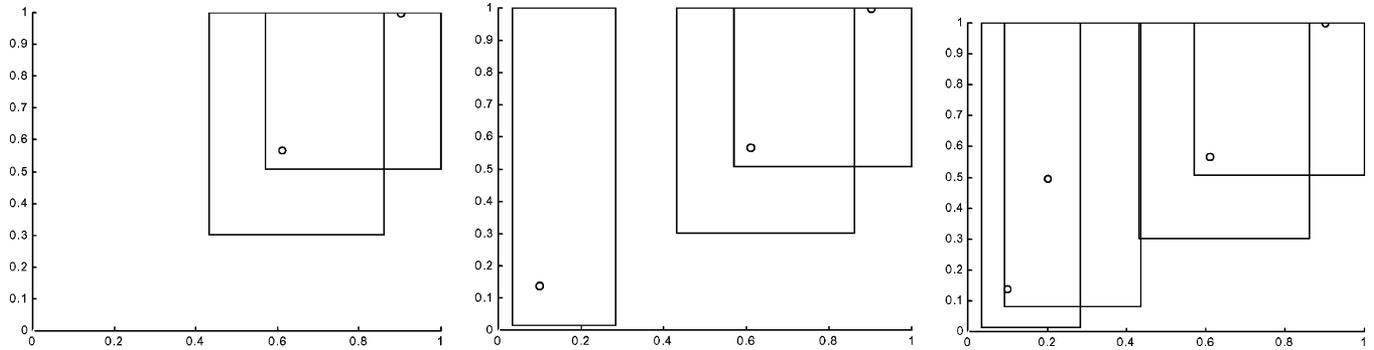


Fig. 17. 2-, 3-, and 4-granular prototypes calculated for data from Example 1.

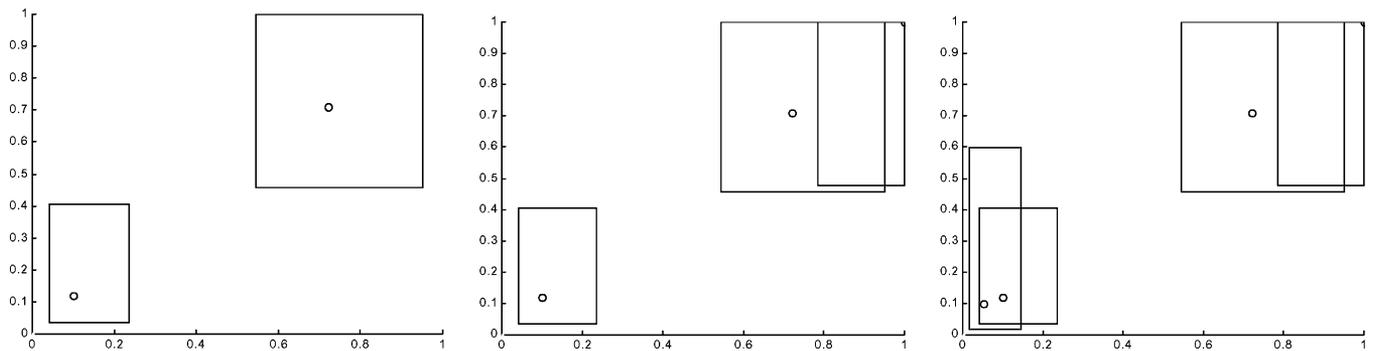


Fig. 18. 2-, 3-, and 4-granular prototypes calculated for data from Example 2.

TABLE V
GRANULAR PROTOTYPES REPRESENTED AS TRIPLES OF LOWER BOUNDS,
MODES (NUMERIC VALUES OF THE PROTOTYPES) AND UPPER BOUNDS (A)
EXAMPLE 1 AND (B) EXAMPLE 2

Prototype 1: {0.431784 0.610000 0.861773} {0.302657 0.570000 1.000000}
Prototype 2: {0.570009 0.900000 1.000000} {0.507569 1.000000 1.000000}
Prototype 3: {0.035332 0.100000 0.283029} {0.015797 0.140000 1.000000}
Prototype 4: {0.091757 0.200000 0.435936} {0.081955 0.500000 1.000000}
(a)
Prototype 1: {0.545043 0.720000 0.951118} {0.457757 0.710000 1.000000}
Prototype 2: {0.042248 0.100000 0.236699} {0.035519 0.120000 0.405423}
Prototype 3: {0.784826 1.000000 1.000000} {0.477658 1.000000 1.000000}
Prototype 4: {0.017296 0.050000 0.144543} {0.016742 0.100000 0.597287}
(b)

performance of the already constructed prototypes. The prototypes themselves are constructed starting from the most “significant” (relevant) so that they come ranked.

- Identification and quantification of possible anisotropy of the feature space: The weight vectors coming with the individual prototypes help quantify the importance of the features. The importance of the features can be local and the ranking the features can vary from prototype to prototype.
- Development of granular prototypes realized on a basis of the clustering results: We showed how the relevance

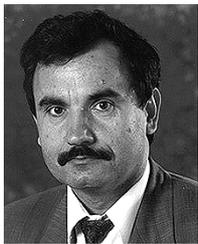
of the prototype can be translated into its granular extension.

These features of the clustering method could be of interest to data analysis. One should stress, however, that the organization of the search for the structure as arranged here could be computationally intensive, especially for large data sets so this method could be considered as a complement to other clustering techniques.

REFERENCES

- [1] M. R. Anderberg, *Cluster Analysis for Applications*. New York: Academic, 1973.
- [2] A. Bargiela, “Interval and ellipsoidal uncertainty models,” in *Granular Computing*, W. Pedrycz, Ed. Heidelberg, Germany: Physica-Verlag, 2001, pp. 23–57.
- [3] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
- [4] B. Bouchon-Meunier, M. Rifqi, and S. Bothorel, “Toward general measures of comparison of objects,” *Fuzzy Sets Syst.*, vol. 84, no. 2, pp. 143–153, 1996.
- [5] A. Di Nola, S. Sessa, W. Pedrycz, and E. Sanchez, *Fuzzy Relational Equations and Their Applications in Knowledge Engineering*. Dordrecht, The Netherlands: Kluwer, 1989.
- [6] M. Delgado, F. Gomez-Skarmeta, and F. Martin, “A fuzzy clustering-based prototyping for fuzzy rule-based modeling,” *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 223–233, 1997.
- [7] M. Delgado, A. F. Gomez-Skarmets, and F. Martin, “A methodology to model fuzzy systems using fuzzy clustering in a rapid-prototyping approach,” *Fuzzy Sets Syst.*, vol. 97, no. 3, pp. 287–302, 1998.
- [8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.
- [9] B. Gabrys and A. Bargiela, “General fuzzy min-max neural network for clustering and classification,” *IEEE Trans. Neural Networks*, vol. 11, pp. 769–783, June 2000.
- [10] F. Hoppner *et al.*, *Fuzzy Cluster Analysis*. Chichester, U.K.: Wiley, 1999.

- [11] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 260–270, Apr. 1995.
- [12] A. Kandel, *Fuzzy Mathematical Techniques with Applications*. Reading, MA: Addison-Wesley, 1986.
- [13] W. Pedrycz, "Direct and inverse problem in comparison of fuzzy data," *Fuzzy Sets Syst.*, vol. 34, pp. 223–236, 1990.
- [14] —, "Neurocomputations in relational systems," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 289–296, Feb. 1991.
- [15] W. Pedrycz and A. Rocha, "Knowledge-based neural networks," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 254–266, 1993.
- [16] W. Pedrycz, *Computational Intelligence: An Introduction*. Boca Raton, FL: CRC Press, 1997.
- [17] —, "Conditional fuzzy clustering in the design of radial basis function neural networks," *IEEE Trans. Neural Networks*, vol. 9, pp. 601–612, June 1998.
- [18] W. Pedrycz and A. V. Vasilakos, "Linguistic models and linguistic modeling," *IEEE Trans. Syst. Man, Cybern.*, vol. 29, pp. 745–757, Dec. 1999.
- [19] W. Pedrycz and A. Bargiela, "Granular clustering: A granular signature of data," *IEEE Trans. Syst., Man, Cybern. B*, vol. 32, pp. 212–224, Feb. 2002.
- [20] P. K. Simpson, "Fuzzy min-max neural networks – Part1: Classification," *IEEE Trans. Neural Networks*, vol. 3, pp. 776–86, Sept. 1992.
- [21] —, "Fuzzy min-max neural networks – Part2: Clustering," *IEEE Trans. Neural Networks*, vol. 4, pp. 32–45, Feb. 1993.
- [22] T. Sudkamp, "Similarity, interpolation, and fuzzy rule construction," *Fuzzy Sets Syst.*, vol. 58, no. 1, pp. 73–86, 1993.
- [23] T. A. Sudkamp and R. J. Hammell II, "Granularity and specificity in fuzzy function approximation," in *Proc. NAFIPS-98*, 1998, pp. 105–109.
- [24] L. A. Zadeh, "Fuzzy sets and information granularity," in *Advances in Fuzzy Set Theory and Applications*, M. M. Gupta, R. K. Ragade, and R. R. Yager, Eds. Amsterdam, The Netherlands: North Holland, 1979, pp. 3–18.
- [25] —, "Fuzzy logic = computing with words," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 2, pp. 103–111, 1996.
- [26] —, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets Syst.*, vol. 90, pp. 111–117, 1997.



Andrzej Bargiela (M'94) is a Professor and Director of the Intelligent Simulation and Modeling Lab at Nottingham Trent University, U.K. His main research focus, pursued since 1978, is processing of uncertainty in the context of modeling and simulation of various physical and engineering systems. The research involves development of algorithms for processing uncertain information, investigation of computer architectures for such processing, and the study of information reduction through visualization. He has published numerous papers and is a coauthor of a research monograph on granular computing (see <http://www.doc.ntu.ac.uk/RTTS>). He serves as Chairman of the European Council of the Society for Computer Simulation (SCS), and is a Member of the U.K. Council of Professors and Heads of Computing. He is also a Member of many program committees at international conferences and serves on the editorial boards of four journals, as well as General Editor of a book series published by the Research Studies Press.



Witold Pedrycz (M'88–SM'94–F'99) is a Professor and Chair with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also a Canada Research Chair (CRC) in Computational Intelligence. He is actively pursuing research in computational intelligence, fuzzy modeling, knowledge discovery and data mining, fuzzy control including fuzzy controllers, pattern recognition, knowledge-based neural networks, relational computation, and software engineering. He has published numerous papers in this area, and is also an author of seven research monographs covering various aspects of computational intelligence and software engineering.

Dr. Pedrycz has been a Member of numerous program committees of IEEE conferences in the areas of fuzzy sets and neurocomputing. He currently serves as an Associate Editor of the *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS* and the *IEEE TRANSACTIONS ON FUZZY SYSTEMS*.



Kaoru Hirota is a Professor and Head of the Department of Computational Intelligence and Systems Science in the Interdisciplinary School of Science and Technology, Tokyo Institute of Technology, Yokohama, Japan. His research interests include fuzzy systems, intelligent robots, image understanding, expert systems, hardware implementations, and multimedia intelligent communication. He was Vice President of the International Fuzzy Systems Association (IFSA) from 1991 to 1993, and is President of the Japan Society for Fuzzy Theory and Systems (SOFT). He is a Senior Associate Editor of the *International Journal of Information Science Applications* and Editor-in-Chief of the *International Journal of Advanced Computational Intelligence*.

Dr. Hirota was an Associate Editor of the *IEEE TRANSACTIONS ON FUZZY SYSTEMS* and the *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*.