Optimised Information Abstraction in Granular Min/Max Clustering

Andrzej Bargiela and Witold Pedrycz

Abstract. The Min/Max classification and clustering has a distinct advantage of generating easily interpretable information granules - represented as hyperboxes in the multi-dimensional feature space of the data. However, while such an information abstraction lends itself to easy interpretation it leaves open the question whether the granules represent well the original data.

In this chapter we discuss an approach to optimised information abstraction, which retains the advantages of Min/Max clustering while providing a basis for building a more representative set of granules. In particular we extend the information density based granulation by including an extra stage of optimised refinement of granular prototypes. The initial granulation is accomplished by creating hyperboxes in the pattern space through the maximisation of the count of data items per unit volume of hyperboxes. The granulation is totally data driven in that it does not make any assumptions about the number or the maximum size of hyperboxes. Subsequent optimisation involves identification of granular prototypes and their refinement so as to achieve full reconstruction of the original data from the prototypes and the corresponding partition matrix.

1 Introductory Comments

Progression from detailed, voluminous numerical information to a more concise representation of knowledge about systems relies to a large extent on an appropriate granulation of information. While it is recognised that the granulation process degrades the accuracy of individual numerical readings the gain of achieving greater generality of models built on such a granulated data is a compelling

Andrzej Bargiela
University of Nottingham, Nottingham, UK
e-mail: andrzej.bargiela@nottingham.ac.uk

Witold Pedrycz
University of Alberta, Edmonton, Canada
e-mail: pedrycz@ee.ualberta.ca

S. Ramanna, L.C. Jain, & R.J. Howlett (Eds.): Emerging Paradigms in ML, SIST 13, pp. 31–48.
springerlink.com

argument for hierarchical structuring of information. Consequently, we observe that information granules permeate most cognitive activities of humans and help organise knowledge for the purpose of decision-making, control, simulation-modelling, prediction, etc. The mathematical frameworks for the generation of information granules have been integrated under the heading of Granular Computing [1-5, 20, 25-28, 30-32] and include most prominently fuzzy sets [12-13, 19, 25], rough sets [17, 30], probabilistic sets as well as crisp sets (interval analysis) [7, 8, 10, 15, 16]. We recognise however, that regardless of the mathematical framework, information granules are conceptual constructs that have to be mediated by the needs of a specific application. In order to reflect the requirements of the real world, i.e. to be anchored in the experimental evidence, information granules need to be easily interpretable in their specific contexts and should have inherent flexibility that allows them to be optimised in some sense [5].

Interestingly, in spite of the large diversity of approaches to information granulation, the majority of clustering techniques published in the literature produce crisp (not granular) prototypes with the corresponding numeric partition matrices. This is surprising because the very rationale of information granulation hinges on the semantical transformation of data from the precise, numeric domain to the graded, granular domain, so the natural expectation would be to have granular representatives of the granular domain.

Nevertheless, some of the previous research has taken a principled view of identifying information granules that represent the granulated data. The min-max clustering, originally proposed in [23] and subsequently developed in [9] adopt a constructive approach to "growing" granular representatives of data assuming a predefined specificity (size in the normalised feature space) of the information granules. In this study we expand on the above approach and adopt a framework of set theory and interval analysis for the granular pre-processing of data and utilise the FCM algorithm to derive the granular representatives of the original data. The granular representatives are then optimised to ensure full representation of the granulated data originating from the pre-processing with the minimum expansion of the prototypes.

Proceeding with a formal description, let us start with a collection of n-dimensional numeric data (located in $\mathbf{R}^n$), say $\boldsymbol{\mathcal{X}} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$. Since we are interested in a data-driven discovery of features rather than the frequency of occurrence of a specific feature we need to ensure that the features of interest are supported by an approximately equal number of input patterns. We also require that the absolute values of readings in individual dimensions in the n-dimensional pattern space do not bias the results. The latter is easily accomplished by normalising the data, i.e. performing a transformation $\mathcal{N}(\boldsymbol{\mathcal{X}}) = \vartheta$ so that $\boldsymbol{\vartheta} = \{\varphi_1, \varphi_2, \ldots, \varphi_N\} \subset [0\ 1]^n$. The second requirement can be expressed formally as follows: given the set of patterns (input set) ϑ of cardinality N and a number of features of interest (clusters) c, we require that each cluster has a support of approximately equal number of patterns. This can be achieved by granulating the original input data, so that one takes full advantage of the detailed information without biasing the subsequent clustering process. The support-balancing granulation process can be expressed formally as $\mathcal{G}(\boldsymbol{\vartheta}) = \Gamma$, where $\boldsymbol{\Gamma} = \{\mathbf{g}_1, \mathbf{g}_2, .., \mathbf{g}_M\}$ and N>M.

The balanced input data set $\mathbf{\Gamma}$ contains both numerical elements (points in $[0\ 1]^n$) and information granules (subsets of $[0\ 1]^n$). A schematic view of the formation of the balanced input set is portrayed in Figure 1.

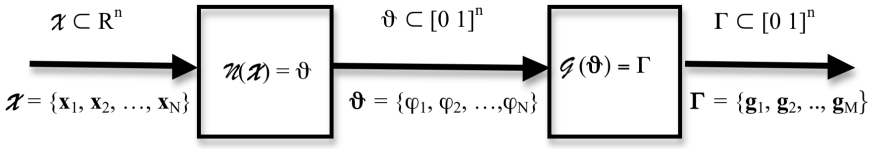

Fig. 1 An overview of the transformation of the numerical data X into support-balancing granules $\tilde{G}$ G(N(***X***) = **G**

Having produced the balanced input data set $\mathbf{\Gamma} = \{\mathbf{g}_1, \mathbf{g}_2, .., \mathbf{g}_M\}$ we may proceed with the construction of *c* information granules $G_1, G_2, \ldots, G_c$ representing the significant features in the input data. Depending upon the formalism of granulation used in clustering algorithm, we can arrive at G_is to be sets, fuzzy sets, rough sets, shadowed sets, etc. [28]. A schematic view of the formation of information granules is portrayed in Figure 2.

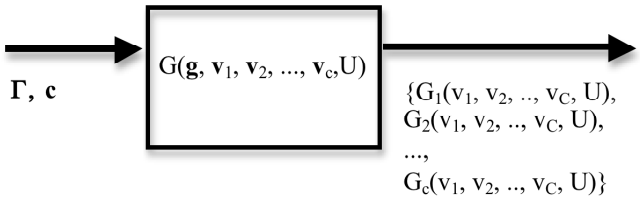

Fig. 2 A general view of clustering of granular data $\mathbf{\Gamma}$ (the granules are represented by prototypes and the partition matrix)

It is worth noting however that unlike in the standard case of clustering of numerical data we are processing here the input set Γ (which contains a mix of numerical and granular entities) thus generating prototypes $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c$ which are themselves information granules. In the case of fuzzy sets, the clusters are characterized by the prototypes and membership functions (grades) forming the corresponding rows of the partition matrix.

The quality of clusters can be assessed via numerous cluster validity indexes [6, 22]. They could be helpful in some cases. They might also play a detrimental role by sending confusing, inconsistent messages as to the most "feasible" number of clusters. This is not surprising as each validity index originates from a certain heuristics being deemed potentially useful in focusing on some selected aspect of the structure of clusters (say, we concentrate on forming clusters as disjoint as possible). Alternatively one can think of the quality of granulation – de-granulation

process in which the composition of the de-granulation and granulation mechanisms should return the result, which is as close to the original entity as possible.

The granulation mechanism returns a representation of any input data (pattern) **g** expressed in terms of membership degrees, $u_1(\mathbf{g})$, $u_2(\mathbf{g})$,…, $u_c(\mathbf{g})$. We use a concise notation $G(\mathbf{g}, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c, U)$ to underline the usage of the mechanism of granulation being applied to **g**. The de-granulation mechanism denoted by G^{-1} applies to the previously obtained result of granulation and returns a certain entity. Ideally, the obtained result should be the same as the original input **g** we have started with. In other words, we require that

$$G^{-1}(G(\mathbf{g}, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c, U))=\mathbf{g} \tag{1}$$

It is well known (for instance, in case of quantization of continuous variables) that the above equality does not hold as we usually encounter $G^{-1}(G(\mathbf{g}, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c, U)) \approx \mathbf{g}$. Given this effect of approximate equality, we can treat a distance between the original **g** and its de-granulated version as a measure of quality of the granulation process (the quality of clusters returned by the clustering technique). In general, we may use the following index

$$\sum_{k=1}^{N} \|G^{-1}(G(\mathbf{g}_k, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c, U))-\mathbf{g}_k\|^2 \tag{2}$$

that can serve as a meaningful indicator of the overall performance of the granulation mechanism. Here the symbol ||. || stands for a certain distance measure. The lower the value of Q is, the better the quality of information granules being assessed in terms of the granulation-de-granulation effect. From the perspective of the concept of information granulation, it is also not surprising that the value of the index is typically non-zero. One could anticipate that the granulation-de-granulation may produce results that are information *granules* rather than single numeric entities. The expectations could be that such information granules produced as a result of this transformation include the original input. In other words, instead of (2), we could expect a satisfaction of the following inclusion

$$\mathbf{g}_k \subset \mathbf{G}_k \tag{3}$$

envisioning that this relationship be (hopefully) satisfied for all (or at least *most*) data $\mathbf{g}_k$. ($\mathbf{G}_k$ represents a de-granulated $\mathbf{g}_k$). The conceptual and ensuing technical question is about the origin of granularity to be used in the formation of the clusters and a way of building it in the representation of the results of clustering.

In this study we will focus on information granules $\mathbf{g}_k$ in the setting of set theory and interval analysis. The rationale behind the selection of this framework is twofold. First, interval representation of granularity leads to a clear interpretation of the results while benefiting from solid mathematical foundations of set theory. Second, the algorithmic layer of set (interval) calculus has been established for a long time and has resulted in a vast number of algorithms [15, 16]. Notwithstanding, the findings reported here can be translated and applied to other frameworks of granular computing such as fuzzy sets (the transltion hinges on the

idea of representing fuzzy sets through their α-cuts [11, 19]; that is, splitting the problem into a family of interval-based granulation tasks).

This chapter is organised as follows. In Section 2, we outline areas in which information granulation plays a crucial role. Section 3, provides a detailed two-level algorithm for information granulation. Summarization of information granules (in terms of identifying granular prototypes) is discussed in Section 4. The verification of the adequacy of granular prototypes is addressed in Section 5.

2 Granular Information in Systems Modeling

There are a number of representative domains where information granules can emerge as a useful vehicle to represent a given problem and make problem solving more efficient [18]. The following three areas are among the most prominent applications of information granulation:

Granulation of Time Series. Time series are commonly encountered in numerous practical problems. There have been various approaches to the description of time series and their classification. They are carried out in the time domain and frequency domain. Prior to any detailed processing, time series are compressed in order to retain the most essential information and suppress details that are deemed redundant from the standpoint of further classification and processing. The essence of granulation of time series is to "discover" dominant components of the series. We may perceive these components as playing a role of basic conceptual blocks easily understood by humans and capturing the semantics of the underlying phenomenon. For instance, information granules may be formed as segments of consecutive samples of the signal. Then each segment may be labeled according to the configuration of the samples, say rapidly increasing signal, steady signal, slowly decreasing signal, etc. Alternatively, as we propose here, one may consider granulating the time series value with its gradient (and/or higher order derivatives) in individual time instances. Note that standard sampling techniques are very specific examples of granulation of time series (as we attempt to capture a segment of a signal falling under a given sampling window by a single numeric value)

Granulation of Digital Images. Digital images are two-dimensional relations. As far as understanding and processing of images is concerned, a crux there is to identify some higher level entities rather than being buried in a minute analysis completed at the level of individual pixels. Such tangible and semantically sound entities are information granules. They may arise at the level of basic homogeneous regions (in terms of brightness, color and texture) one can identify in an image. These entities are inherently hierarchical: at a higher level we may think of individual objects in the image (that are composed of the granules arising at the lower level with more specific and less abstract information granules). At the technical end, the simplest and least abstract information granules are formed by defining n-by-m blocks of pixels. At the higher level, we are concerned with various clustering techniques that help us construct abstractions out of the low-end (more detailed) information granules such as the already mentioned blocks of pixels.

Granulation of Spatial Structures. An array of current modeling pursuits occurs in the realm of distributed systems such as networks (both physical and virtual). Obvious examples of these architectures are public utility networks, telecommunication networks, social networks, supply chain networks, etc. In spite of their evident diversity, the networks share several profound commonalties. In particular, a hierarchical type of modeling is omnipresent there. Instead of analyzing the entire network, we split it into subnetworks (modules) that are loosely connected and proceed with a detailed analysis at this level. Obviously, this task is more tangible and manageable from the computational and interpretation standpoint. Each subnetwork is an information granule that is afterwards treated as a conceptual and algorithmic entity. For instance, when looking into a flow of traffic in a complex network, we partition the network into modules (call them telecommunications granules) and study all incoming and outgoing traffic from this perspective. The concept of hierarchy and information granulation is inherently associated with GIS (Geographic Information Systems) systems where we anticipate various levels of detail and control the process of concentrating on specific aspects by establishing proper levels of information granularity.

3 Information Density Based Granulation

In this section we focus on the algorithmic layer of the transformation of the numerical data set $\boldsymbol{\mathcal{X}}$ into a balanced granular data set $\boldsymbol{\Gamma}$, as illustrated in Figure 1. The normalisation stage, $\mathcal{N}(\boldsymbol{\mathcal{X}}) = \vartheta$, transforming input set $\boldsymbol{\mathcal{X}} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subset R^n$ into $\boldsymbol{\vartheta} = \{\varphi_1, \varphi_2, \ldots, \varphi_N\} \subset [0\ 1]^n$ is straightforward and can be expressed for individual input data as:

$$\vartheta_i = \mathbf{N}\,\mathbf{x}_i - \mathbf{o} \tag{3}$$

where $\mathbf{N} \in \mathbf{R}^{DxD}$ is a normalisation matrix

$$\mathbf{N} = \begin{bmatrix} \dfrac{1}{\max\limits_i x_{i,1} - \min\limits_i x_{i,1}} & 0 & . & 0 \\ 0 & \dfrac{1}{\max\limits_i x_{i,2} - \min\limits_i x_{i,2}} & . & 0 \\ . & . & . & . \\ 0 & 0 & . & \dfrac{1}{\max\limits_i x_{i,D} - \min\limits_i x_{i,D}} \end{bmatrix}$$

and $\mathbf{o} \in \mathbf{R}^{Dx1}$ is an offset vector constructed as

$$\mathbf{o} = \begin{bmatrix} \dfrac{\min\limits_i x_{i,1}}{\max\limits_i x_{i,1} - \min\limits_i x_{i,1}} \\ \dfrac{\min\limits_i x_{i,2}}{\max\limits_i x_{i,2} - \min\limits_i x_{i,2}} \\ \vdots \\ \dfrac{\min\limits_i x_{i,D}}{\max\limits_i x_{i,D} - \min\limits_i x_{i,D}} \end{bmatrix}$$

with max- and min- operations performed over all data items i $\in \{1, 2, \ldots, N\}$ in the D-dimensional pattern space.

The support-balancing granulation $\mathcal{G}(\vartheta) = \Gamma$ arises as a compromise between two conflicting requirements:

1. Each granule $\mathbf{g}_i$ should embrace as many elements of ϑ as possible (to be a sound representation of the underlying data). This can be expressed formally as maximising the cardinality of the $\mathbf{g}_i$ set.
2. The granule should be highly specific (its size in every dimension in the multi-dimensional pattern space should be as small as possible). The size could be measured in each dimension as a simple interval but it is convenient to represent this in a slightly more general way as function defined on such intervals (this is to avoid numerical problems with intervals of zero-length).

The algorithm implementing such a granulation has been proposed in [2] and it is briefly outlined here. The granulation is implemented as a one-pass process:

1. Initialise data structures representing cardinality and the width of individual data items (1 and 0, respectively for the point-data)
2. Calculate and store the values of "information density" (the ratio of the cardinality of the granule and the functional measurement of its size) of hypothetical granules formed by any two data items in the input data set. This forms an upper-diagonal matrix $\boldsymbol{D}$ of size NxN, where N is the cardinality of the input data set.
3. Find the maximum entry in $\boldsymbol{D}$.
4. If the maximum corresponds to an off-diagonal element (i-th and j-th coord):

- merge the two information items (identified by the i-th row and j-th column) into a single information granule, which has width defined by the maximum and minimum values of coordinates in each dimension from the two component granules; i.e.:
    - find the min of each of the coordinates of the i-th and j-th granule;
    - find the max of each of the coordinates of the i-th and j-th granule;
    - modify the i-th granule so that its size is defined by min- and max- values identified above;

- update the cardinality of the resulting granule to the sum of the cardinality counts of the component granules;
- update the i-th row and column of ***D*** with the information pertinent to the newly formed information granule and remove the j-th row and column from ***D***; i.e.:
    - copy rows 1 to j-1 from matrix ***D*** to matrix ***D1***;
    - copy rows j+1 to size(***D***) from matrix ***D*** as rows j to size(***D***)-1 in martix ***D1***;
    - copy columns 1 to j-1 from matrix ***D***1 to matrix ***D2***;
    - copy columns j+1 to size(***D***) from matrix ***D1*** as columns j to size(***D***)-1 in martix ***D2***;
    - overwrite matrix ***D*** with matrix ***D2***;
- return to 3)

5. If the maximum corresponds to a diagonal element (i=j):

- copy the granule to an output list and remove the corresponding row and column from matrix ***D***; i.e.:
    - copy rows 1 to j-1 from matrix ***D*** to matrix ***D1***;
    - copy rows j+1 to size(***D***) from matrix ***D*** as rows j to size(***D***)-1 in martix ***D1***;
    - copy columns 1 to j-1 from matrix ***D1*** to matrix ***D2***;
    - copy columns j+1 to size(***D***) from matrix ***D1*** as columns j to size(***D***)-1 in martix ***D2***;
    - overwrite matrix ***D*** with matrix ***D2***;

6. If the size of matrix ***D*** is greater than 1, return to 3), otherwise terminate.

Computational complexity of this granulation algorithm is $O(N^2)$ owing to the computations of matrix ***D*** in step 2). However, unlike the clustering techniques (such as FCM), the granulation process has an inherently local character and can be easily applied to a partitioned input data thus circumventing the high computational cost associated with large data sets. It is worth pointing out that the size of matrix ***D*** is being reduced by one row and column at each iteration thus the number of iterative steps equals N-1. Since the algorithm maintains linear computational complexity with respect of the input space dimension (not to be confused with the complexity with respect of the cardinality of the data set which is $O(N^2)$), it is particularly suitable for processing multi-dimensional data. Also, it is worth pointing that the algorithm maintains a localized view of data. As the granulation proceeds, the identified granules do not exercise further influence on data points that remain after their removal.

It is worth noting that unlike the standard Min-Max [9, 23] the above algorithm does not make any assumptions about the maximum size of granules. Granules are allowed to grow as long as their local data density keeps increasing. Furthermore, there are no assumptions about the separation of cluster centres. The formation of closely separated granules is largely avoided by the very nature of maximisation of information density, which tends to increase the size of granule if it means

adding sufficiently large number of data items (another granule) without undue increase of its volume. If, on the other hand, the increase in volume would imply the reduction of information density, the granule does not expand and remains well separated from the neighboring granules. Another distinguishing feature of our algorithm is that it allows processing both point-size and hyperbox data. This is an important characteristic that allows hierarchical granulation of data. It should be noted that hierarchical granulation enables overcoming the limitations of the 'local view' of data while supporting the application of the algorithm to a partitioned input data set.

To illustrate the operation of the granulation algorithm we consider a synthetic data set shown in Figure 3. The data set comprises of 100 data items in a large cluster and 10 data items in a small cluster. The granulation algorithm produces a more balanced granular data set comprising of 16 information granules; with 14 granules representing the large cluster and 2 granules representing the small cluster. Subsequent application of the algorithm to this granular data produces a further reduction of the count of granules to 5, 3 and 2 at level-2, level-3, and level-4 granulation respectively. In the same time the ratio of data items contained in the large and the small cluster is reduced from 100/10, 14/2, 4/1, 2/1, 1/1 (Figure 4a-4d).

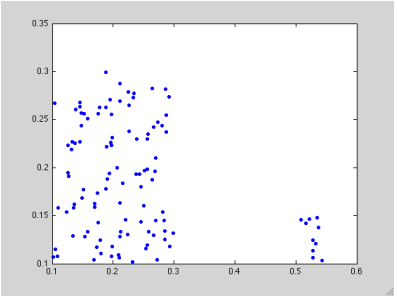

Fig. 3 Synthetic data set with two data clusters with 100 and 10 data items

It is worth noting that the number of granulation levels does not need to be defined in advance. The hierarchical granulation is simply carried out until the number of granules identified at the subsequent granulation levels does not change. Of course, in any practical application the maximum size of granules is frequently pre-defined so that the granules map well onto some linguistic entities.

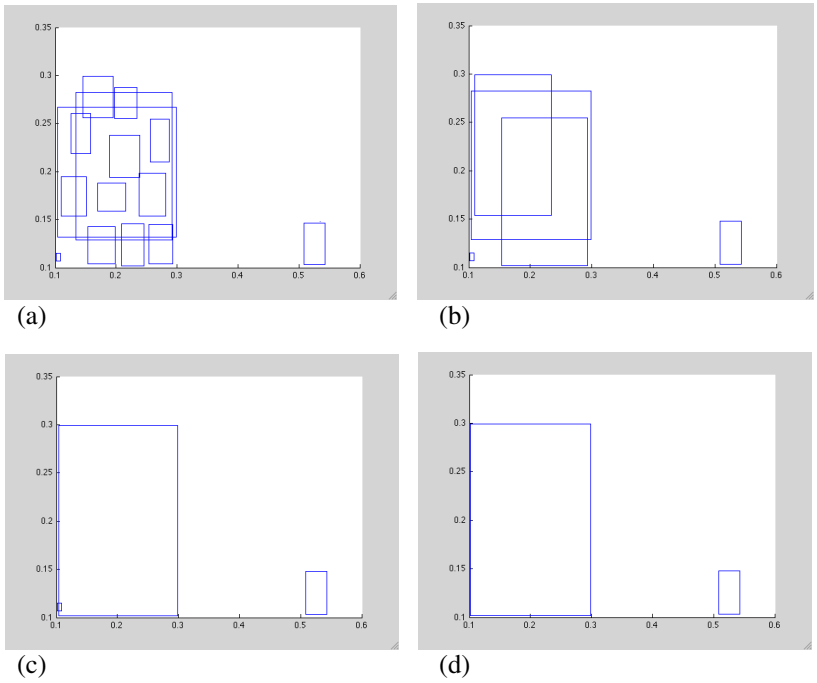

Fig. 4 (a) Information granules produced by the algorithm applied to the original data (16 granules); (b) Level-two information granules (5 granules); (c) Level-three information granules (3 granules); and (d) Level-four information granules (2 granules).

4 Granular Representatives of Data

The recursive application of the granulation algorithm discussed in the previous section, condense the data quite significantly. What is of fundamental interest though is whether this 'condensing' preserves the essential characteristics of data. We assess here the ability to preserve the essential characteristics of data by identifying a limited number of representatives of both the original numeric data **x** and the constructed information granules **g**. This is accomplished by clustering and identifying prototypes (representatives) of the original data and the granules, cf [7, 8]. In particular, we adopt a fuzzy clustering method - a well-known FCM algorithm [6]. It is instructive to recall briefly the formal description of the FCM algorithm so as to appreciate the nature of the optimisation problem represented in Figure 2. The granulation of **g** returns its representation through the collection of available information granules expressed in terms of their prototypes. More specifically, **g** is expressed in the form of the membership grades of **g** to the individual granules G_i, which form a solution to the following optimisation problem

$$\text{Min} \sum_{i=1}^{c} u_i^m(\boldsymbol{g}) \| \boldsymbol{g} - \mathbf{v}_i \|^2$$

subject to constraints

$$\sum_{i=1}^{c} u_i(\boldsymbol{g}) = 1 \quad u_i(\boldsymbol{g}) \in [0,1] \tag{4}$$

where "m" stands for the fuzzification coefficient, m > 1. We note however that, unlike in the original FCM formulation, the data **g** is not exclusively numeric (i.e. $\mathbf{g} \notin \mathbf{R}^n$) but involves information granules represented as hyperboxes in $\mathbf{R}^n$. In other words $\mathbf{g}=[\mathbf{g}^-, \mathbf{g}^+]$ where $\mathbf{g}^-$ are the min-value coordinates and $\mathbf{g}^+$ are the max-value coordinates of a hyperbox **g**. Since both $\mathbf{g}^-, \mathbf{g}^+ \in \mathbf{R}^n$ by concatenating the min- and max-coordinates of the hyperbox we can represent it as $\mathbf{g} \in \mathbf{R}^{2n}$. In this expanded space any numerical data **x** is represented as $\mathbf{g}=[\mathbf{x}^-, \mathbf{x}^+]$, where $\mathbf{x}=\mathbf{x}^-=\mathbf{x}^+$. The distance $\| \boldsymbol{g} - \mathbf{v}_i \|^2$ can be interpreted as the sum of distances between the minimum- and maximum-value coordinates of the respective hyperboxes, as illustrated in Figure 5.

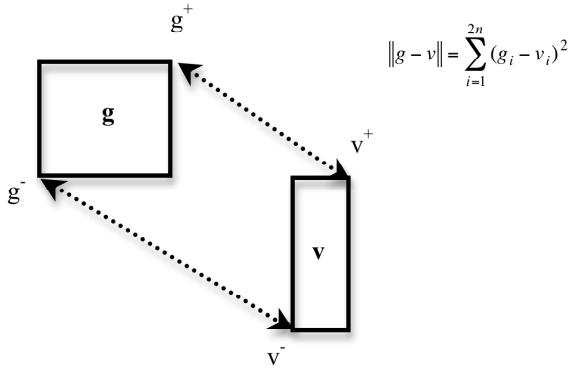

Fig. 5 Distance between hyperboxes.

Consequently the FCM solution derived in the augmented data space ($\mathbf{R}^{2n}$) reads as follows

$$u_i(\boldsymbol{g}) = \frac{1}{\sum_{j=1}^{c} \left(\frac{\| \boldsymbol{g} - \mathbf{v}_i \|}{\| \boldsymbol{g} - \mathbf{v}_j \|} \right)^{2/(m-1)}} \tag{5}$$

Applying the FCM algorithm to the original and to the granulated data we identify prototypes **v** (illustrated in Figure 6 and 7) and the partition matrices u(**g**). It is clear that the prototypes evaluated for the original data are biased towards the more numerous data in the large cluster. Although we have specified 2 and 3

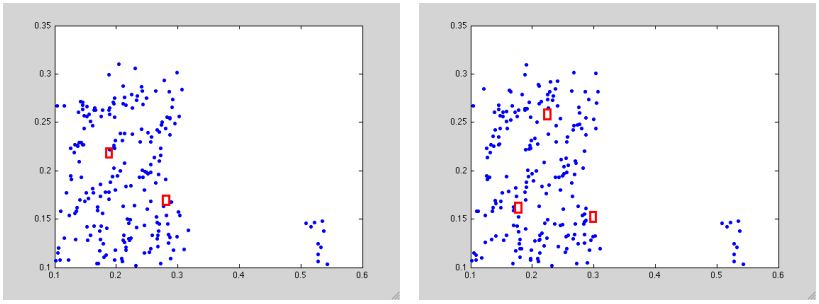

Fig. 6 FCM prototypes found for the original numeric data; (a) two prototypes; (b) three prototypes; (the prototypes are numerical but for the sake of clarity are represented here as rectangles). Note that none of the prototypes is positioned within the small data cluster

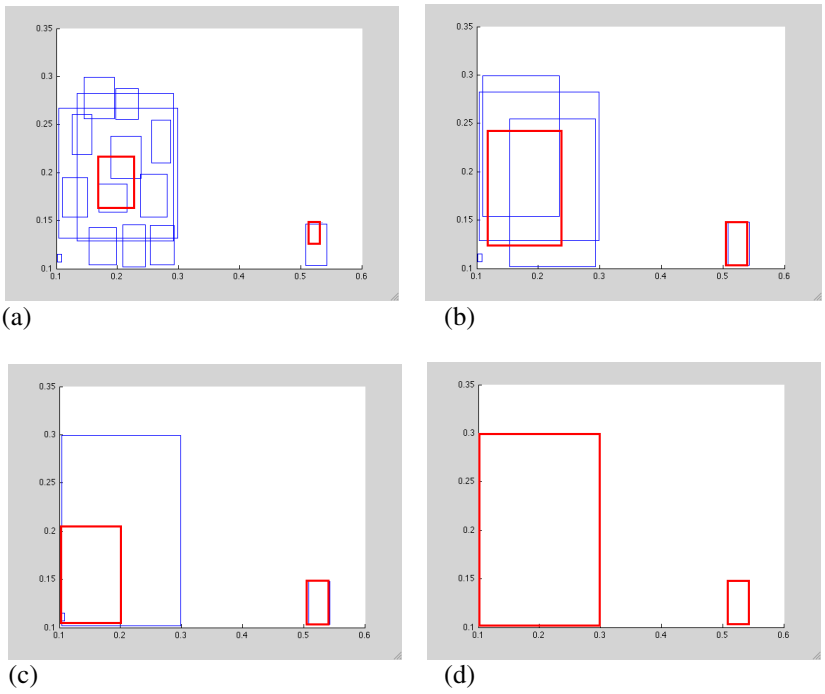

Fig. 7 FCM prototypes found for the balanced, granulated data; (a) level-one; (b) level-two; (c) level-three; (d) level-four granulation; Note that for all granulation levels the small cluster of data is represented by a granule.

prototypes as a target for the FCM calculations (which should be sufficient to characterize the two clusters), the small cluster of data appears to be overwhelmed by the sheer numbers of data points in the large cluster. This is an unfortunate effect because the lack of representation of the small cluster by a local prototype is certain to lead to subsequent misclassifications of data.

By contrast, the FCM prototypes derived for the granulated data (Figure 7) capture the presence of two data clusters at all levels of data granulation by associating one prototype with each data cluster. However, the size of the granular prototypes varies depending on the level of granulation of the original data. This suggests that there is a need to develop a criterion for the selection of the appropriate level of granularity of the prototypes. The actual values of min- and max-coordinates for the various levels of data granulation are given in Table 1.

Table 1 FCM Prototypes for different levels of data granulation

Granulation Level	**Min/Max coordinates of prototypes**			
	$\mathbf{v}^-_1$	$\mathbf{v}^-_2$	$\mathbf{v}^+_1$	$\mathbf{v}^+_2$
Level 0; original numeric data (Fig. 6a)	0.1865 0.2708	0.2203 0.1580	0.1865 0.2708	0.2203 0.1580
Level 1; granulated data (Fig. 7a)	0.5127 0.1690	0.1261 0.1634	0.5309 0.2275	0.1483 0.2166
Level 2; granulated data (Fig. 7b)	0.1178 0.5046	0.1235 0.1034	0.2378 0.5392	0.2421 0.1475
Level 3; granulated data (Fig. 7c)	0.5054 0.1027	0.1034 0.1045	0.5408 0.2011	0.1483 0.2047
Level 4; granulated data (Fig. 7d)	0.5082 0.1016	0.1034 0.1020	0.5429 0.2988	0.1476 0.2995

One possible approach to the assessment of the quality of the FCM solution is to measure its ability to represent the majority of the original data, [21]. This can be accomplished by reconstructing original $\mathbf{g}$ using the prototypes $\mathbf{v}$ and the membership grades $u_i(\mathbf{g})$. It should be noted that although $\mathbf{g}$ represents the granulated data, it can also be considered as a representation of the original numeric data where each data point $\mathbf{x} \in \mathbf{R}^n$ is seen as a granule $\mathbf{g}=[\mathbf{x},\mathbf{x}] \in \mathbf{R}^{2n}$. The data reconstruction task can be formally expressed as evaluation of the vector $\hat{g}$ through a solution to the minimization problem

$$\sum_{i=1}^{c} u_i^m(\boldsymbol{g}) \| \hat{\boldsymbol{g}} - \mathbf{v}_i \|^2 \tag{6}$$

Because of the use of the Euclidean distance, the calculations here are straightforward yielding the result

$$\hat{\boldsymbol{g}} = \frac{\sum_{i=1}^{c} u_i^m(\boldsymbol{g})\mathbf{v}_i}{\sum_{i=1}^{c} u_i^m(\boldsymbol{g})} \tag{7}$$

5 Granular Refinement of Prototypes

Granular prototypes **v** obtained through the application of the FCM are dependent on the level of granulation of the original data (including the case where no data-balancing granulation has been attempted). In order to free the prototypes from this dependency, we refine the structure of the prototypes by admitting a variation of their granularity

$$V_{ij} = [\mathbf{v}_{ij}^{\min} - \sum \text{range}_j, \mathbf{v}_{ij}^{\max} + \sum \text{range}_j] \tag{8}$$

i=1, 2,…, c, j=1, 2, …,n and $V_{ij} \in [0\ 1]^n$. Note that each prototype increases its *granularity* to the same extent with regard to all variables. Both min-coordinates $\mathbf{v}_{ij}^{\min}$ and max-coordinates $\mathbf{v}_{ij}^{\max}$ of the prototype are transformed so as to produce a symmetrical enlargement of v_{ij} by the imposed level of granularity $\sum$.

The main limitation of this construction is that all variables are treated in the same way by assigning to all of them the same value of $\sum$. We attempt to offset some of this limitation by using a granule-specific multiplier *range* that promotes the preservation of topological similarity between the original and the modified prototypes. However, if the modified granule extends outside the unit hyperbox, we enforce the requirement that $V_{ij} \in [0\ 1]^n$ thus, in effect we generate a non symmetrical expansion of the prototype v_{ij}. We denote the resulting prototype-specific expansion factors as $\tilde{\varepsilon}_i$.

As the granularity is sought of as an important modeling resource to be prudently allocated, its distribution needs more attention. We propose here that the sum of all $\tilde{\varepsilon}_i$ is minimized. In other words we establish a performance criterion as

$$Q = \min \sum_{i=1}^{c}\sum_{j=1}^{n} \tilde{\varepsilon}_{ij} \tag{9}$$

where $\tilde{\varepsilon} = [\tilde{\varepsilon}_1, \tilde{\varepsilon}_2, \ldots, \tilde{\varepsilon}_c]$, $\tilde{\varepsilon}_i = [\tilde{\varepsilon}_{i1}, \tilde{\varepsilon}_{i2}, \ldots, \tilde{\varepsilon}_{in}]$. We require that every original data item g_k is enclosed in the reconstructed data item represented by a hyperbox $\hat{g}_k$ evaluated as in (7). We can express this formally as

$$\text{card}\{\boldsymbol{g}_k \in G^{-1}(G(\boldsymbol{g}_k, \mathbf{V}_1(\tilde{\varepsilon}_1), \mathbf{V}_2(\tilde{\varepsilon}_2), \ldots, \mathbf{V}_c(\tilde{\varepsilon}_c), U)\} = N \tag{10}$$

The constraint (10) represents the most stringent requirement on the reconstruction of data and may in some cases be replaced by a less stringent requirement, that a given proportion of data is correctly reconstructed. Evidently, the high values of $\tilde{\varepsilon}_i$ are more likely to satisfy (10) but the resulting prototypes would imply lack of

specificity and might not be acceptable in many applications. In other words, the performance index (9) captures the nature of consistent and parsimonious granular representation of data through clustering: we strive to achieve a situation where all patterns when being represented by the granular prototypes are positioned within the bounds resulting through the reconstruction process (7). By noting an indirect way in which the constraint (10) depends upon the vector of variables $\tilde{\varepsilon}_i$ we resort to the use of a population–based optimisation method. One of the viable alternatives is the Particle Swarm Optimisation (PSO), see [24].

A particle swarm is a population of particles representing possible solutions located in the multidimensional search space [14, 24, 29]. Each particle explores the search space and during this search it adheres to some quite intuitively appealing guidelines navigating the search process: (a) it tries to follow its previous direction, and (b) it looks back at the best performance recorded so far both at the level of the individual particle as well as the entire population.

The algorithm exhibits some societal aspects of interaction. There is some collective search of the problem space along with some component of memory incorporated as an integral part of the search mechanism. The performance of each particle during its traversal of the search space is assessed by means of some performance index (fitness function). A position of a swarm in some search space **S** of a given dimensionality "r", is described by some vector **z**(t)**S** where "t" denotes consecutive discrete time moments (generation index). The method can be briefly explained as follows. The next position of the particle is governed by the following update expressions concerning the particle, **z**(t+1) and its speed, **v**(t+1)

$$\begin{aligned} z_i(t+1) &= z_i(t) + v_i(t+1) && \text{//update of position of the particle} \\ v_i(t+1) &= \xi\, v_i(t) + f_{1i}(p_i - z_i(t)) + f_{2i}\,(p_{total,i} - z_i(t)) && \text{// update of speed of the particle} \end{aligned} \tag{11}$$

i=1, 2, …, r where **p** denotes the best position (characterized by the lowest performance index) reported so far for this particle, $\mathbf{p}_{total}$ is the best position overall developed so far across the entire population. The two other parameters of the PSO that is f_{1i} and f_{2i} are random numbers drawn from the uniform probability distribution defined over the [0,2] interval that help build a proper combination of the components of the speed; different random numbers affect the individual coordinates of the speed.

The second expression governing the change in the velocity of the particle is particularly interesting as it captures the relationships between the particle and its history as well as the history of the overall population in terms of their performance reported so far. The current speed **v**(t) is impacted by the inertial weight (ξ) smaller than 1 whose role is to articulate some factor of resistance to change the current speed (the values of the inertia weight are kept below 1). The fitness function is given by (9). The particle consists of the coordinates of $\tilde{\varepsilon}_i$. Before it is used to evaluate the fitness function, the coordinates are normalized so that they satisfy the requirement expressed by (10).

Application of PSO to the prototypes reported in Table 1 produced optimised prototypes at each level of data granulation. The results detailed in Table 2 and Figure 8 suggest that the Level-2 granulation produces the best results in terms of interpretability and representativeness of granular prototypes of data.

Table 2 Refined prototypes for different levels of data granulation with the corresponding value of the performance index.

Granulation Level	**Coordinates of refined prototypes**				**Q**
	v^-_1	v^-_2	v^+_1	v^+_2	
Level 0;	0	0	0.4902	0.5241	2.0509
numeric data (Fig. 8a)	0	0	0.5735	0.4632	
Level 1;	0.4157	0.0291	0.6279	0.2453	0.7760
granulated data (Fig. 8b)	0.0720	0.0664	0.3244	0.3136	
Level 2;	0.4456	0.0444	0.5982	0.2065	**0.4720**
granulated data (Fig. 8c)	0.0588	0.0645	0.2968	0.3011	
Level 3;	0.4064	0.0044	0.6398	0.2473	0.7538
granulated data (Fig. 8d)	0.0037	0.0055	0.3002	0.3038	

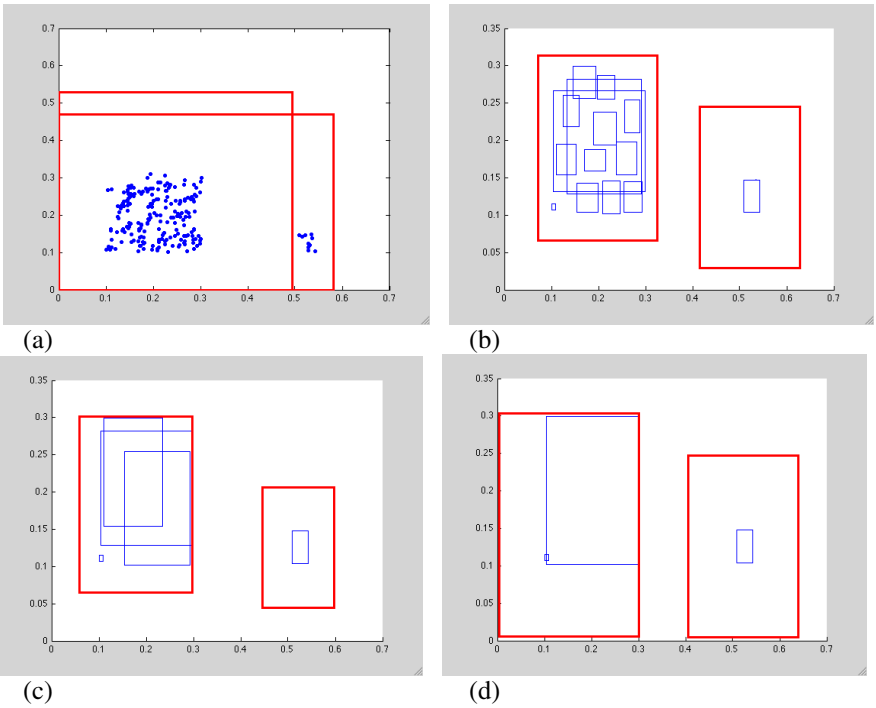

Fig. 8 FCM prototypes refined by a granular expansion; (a) Original numeric data; (b) Level-one; (c) Level-two; (d) Level-three granulated data. The prototypes satisfy the full reconstruction requirement (10).

6 Conclusions

Aggregation of detailed numerical information into information granules promotes a more global view of data and of the application domain from which the data is derived. However, the effectiveness of this information abstraction depends on the quality of the resulting information granules. This study focuses on two essential characteristics of information abstraction: the interpretability and the representativeness of information granules. By adopting the min/max granulation framework we generate highly interpretable hyperboxes in the pattern space while accepting the expense of having to use a greater number of hyperboxes to represent some of the more complex topologies of data clusters. The increasing levels of generalization of data lead naturally to a hierarchical structure of the proposed granulation method. The proposed evaluation and subsequent refinement of granular prototypes – representing the information granules at each level of abstraction – offers a means of maximizing the expressive power of information granules by ensuring that there is no loss of information in the granulation-degranulation process.

The study opens a large spectrum of possibilities for the refinement of the balance between the interpretability and the representativeness of information granules. In particular, some applications might be tolerant of some loss of the ability to represent fully the original data if this meant the enhancement of the specificity (small size) of the prototypes. Conversely, other applications may depend critically on the ability of the prototypes to represent all input data. The small synthetic data set discussed here is intended to promote such considerations as opposed to providing a reference solution for a specific application.

References

1. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Dordrecht (2003)
2. Bargiela, A., Pedrycz, W.: Recursive information granulation: Aggregation and interpretation issues. IEEE Trans. on Syst. Man and Cybernetics 33(1), 96–112 (2003)
3. Bargiela, A., Pedrycz, W.: Granular mappings. IEEE Transactions on Systems, Man, and Cybernetics-part A 35(2), 292–297 (2005)
4. Bargiela, A., Pedrycz, W.: A model of granular data: a design problem with the Tchebyschev FCM. Soft Computing 9, 155–163 (2005)
5. Bargiela, A., Pedrycz, W.: Toward a theory of Granular Computing for human-centered information processing. IEEE Transactions on Fuzzy Systems 16(2), 320–330 (2008)
6. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, N. York (1981)
7. Chiu, S.: Method and software for extracting fuzzy classification rules by subtractive clustering. In: NAFIPS, pp. 461–465 (1996)
8. Cios, K., Pedrycz, W., Swiniarski, R.: Data Mining Techniques. Kluwer Academic Publishers, Boston (1998)
9. Gabrys, B., Bargiela, A.: General fuzzy min-max neural network for clustering and classification. IEEE Trans. on Neural Networks 11(3), 769–783 (2000)
10. Hata, Y., Mukaidono, M.: On some classes of fuzzy information granularity and their representations. In: ISMVL 1999, Japan, pp. 288–293 (1999)

11. Kandel, A.: Fuzzy Mathematical Techniques with Applications. Addison-Wesley, Reading, MA (1986)
12. Kacprzyk, J., Yager, R.R.: Linguistic summaries of data using fuzzy logic. Int. J. General Systems 30, 33–154 (2001)
13. Kacprzyk, J., Zadrozny: Linguistic database summaries and their protoforms: toward natural language based knowledge discovery tools. Information Sciences 173, 281–304 (2005)
14. Ling, S.H., Iu, H.H.C., Chan, K.Y., Lam, H.K., Yeung, B.C.W., Leung, F.H.: Hybrid Particle Swarm Optimization with wavelet mutation and its industrial applications. IEEE Transactions on Systems, Man, and Cybernetics, Part B 38(3), 743–763 (2008)
15. Moore, R.E.: Interval Analysis. Prentice Hall, Englewood Cliffs (1966)
16. Kreinovich, V., Lakeyev, A., Rohn, J., Kahl, P.: Computational Complexity and Feasibility of Data Processing and Interval Computations. Kluwer, Dordrecht (1998)
17. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic, Dordrecht (1991)
18. Pedrycz, W.: Computational Intelligence: An Introduction. CRC Press, Boca Raton (1997)
19. Pedrycz, W., Gomide, F.: An Introduction to Fuzzy Sets. MIT Press, Cambridge (1998)
20. Pedrycz, W., Bargiela, A.: Information granulation: A search for data structures. In: Knowledge-based Engineering Systems KES 2001, Osaka, pp. 1147–1151 (October 2001)
21. Pedrycz, W., Valente de Oliveira, J.: A development of fuzzy encoding and decoding through fuzzy clustering. IEEE Transactions on Instrumentation and Measurement 57(4), 829–837 (2008)
22. Pedrycz, W.: Knowledge-Based Fuzzy Clustering. John Wiley, N. York (2005)
23. Simpson, P.K.: Fuzzy min-max neural networks. IEEE Transactions on Neural Networks 3, 776–786 (1992)
24. Van den Bergh, F., Engelbrecht, A.P.: A study of particle swarm optimization particle trajectories. Information Sciences 176(8), 937–971 (2006)
25. Zadeh, L.A.: Fuzzy sets and information granularity. In: Gupta, M.M., Ragade, R.K., Yager, R.R. (eds.) Advances in Fuzzy Set Theory and Applications, pp. 3–18. North Holland, Amsterdam (1979)
26. Zadeh, L.A.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems 90, 111–117 (1997)
27. Zadeh, L.A.: From computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions. IEEE Trans. on Circuits and Systems 45, 105–119 (1999)
28. Zadeh, L.A.: Toward a generalized theory of uncertainty (GTU) – an outline. Information Sciences 172(1-2), 1–40 (2005)
29. Zhan, Z., Zhang, J., Li, Y., Chung, H.S.H.: Adaptive Particle Swarm optimization. IEEE Trans. on Systems, Man, and Cybernetics, Part B 39(6), 1362–1381 (2009)
30. Yao, Y.Y.: Information granulation and rough set approximation. International Journal of Intelligent Systems 16(1), 87–104 (2001)
31. Yao, Y.: A unified framework of granular computing. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing, pp. 401–410. Wiley-Interscience, New York (2008)
32. Yao, Y.Y.: Integrative levels of granularity. In: Bargiela, A., Pedrycz, W. (eds.) Human Centric Information Processing Through Granular Modelling, pp. 31–47. Springer, Berlin (2009)