

Granular classifiers and their design through refinement of information granules

Abdullah Balamash³  · Witold Pedrycz^{1,2,3} · Rami Al-Hmouz³ · Ali Morfeq³

Published online: 30 December 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract In this study, we focus on the design and refinements of granular pattern classifiers, namely classifiers, which deal with a collection of information granules formed in a certain feature space. The development of this category of classifiers is realized as a two-phase design process. First, information granules occupying some regions of the feature space are formed through invoking mechanisms of clustering or fuzzy clustering. As a result, regions in the feature space are built, which are densely occupied by the patterns predominantly belonging to the same class. We offer a detailed way of assessing the character and quality of information granules and their information (classification-oriented) content. The resulting description is utilized in the realization of the classification mechanism being considered at the second phase of the design of the granular classifier. The mapping from the collection of information granules to class assignment (classification) involves matching of a pattern to be classified to individual information granules and aggregating them by considering the information content of the corresponding granules. In the study, a number of descriptors capturing information content and aggregation functions are analyzed. To improve the performance of the granular clas-

sifier, a refinement of information granules is carried out, in which highly heterogeneous information granules (viz. those containing patterns belonging to various classes) are refined (split, specialized), and their refined versions are afterwards used in the buildup of the classifier. A series of experiments involving both synthetic data as well as those publicly available is reported and analyzed, illustrating the main advantages of granular classifiers and their design procedure.

Keywords Granular computing · Information granules · Fuzzy clustering · Successive refinements of clusters · Information content · Diversity

1 Introduction

Information granularity is a fundamental concept associated with an abstract view of phenomena and, as such, permeating a human way of perceiving the world, acquiring and organizing knowledge, realizing reasoning processes, and communicating findings. Information granules are the operational constructs involved there. Granular computing (Apolloni et al. 2008; Pedrycz 2013) has emerged as a discipline concerned with acquisition, processing, and interpretation of information granules. In pattern recognition and classification problems in particular, information granularity is evidently visible. On a basis of experimental data, we construct classifiers, viz. form mappings that discriminate between patterns belonging to different classes. Granular classifiers form a category of classifiers whose design and function revolve around information granules built in the feature space. The design process comprises two phases. First, information granules are formed with the anticipation that they help establish some homogeneous regions in the fea-

Communicated by V. Loia.

✉ Abdullah Balamash
asbalamesh@kau.edu.sa

¹ Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada

² Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

³ Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

ture space, viz. composed of patterns belonging to a single class. While there are different ways to build the information granules (Al-Hmouz et al. 2015, 2014), the focus here is to apply the expansion idea as it was presented in Balamash et al. (2015), to reduce the diversity within these information granules to improve the classification performance. The second design phase is about forming a sound mechanism of aggregating levels of matching incoming patterns with the information granules in the feature space and aggregating partial results by taking into account the content of individual information granules.

The main idea is to first construct a collection of information granules at a high level of abstraction and then, as needed, try to refine these information granules to form more detailed ones. There are two fundamental concepts that are behind the formation of the granular classifiers, namely (1) the classification content of information granules and (2) the refinement of information granules. The refinement is carried out by expanding some of the initial information granules and considering criteria that can maximize the regression or classification performance.

The selection of the cluster (information granule) to refine (specialize) is a key design question. In the case of regression, the diversity of the output associated with the information granule entities was used and was found to be a good choice (Balamash et al. 2015). For the classification problem, a given information granule represents each classification class with a certain degree, and accordingly, the total misrepresentation the information granule has, with respect to all its entities, is a sound choice for deciding about the candidate information granule to refine.

In essence, the way of designing the granular classifier presented in the study follows the idea of the refined regression model presented in Balamash et al. (2015), where we demonstrated the applicability of information granules in building regression models.

This paper is structured as follows. In Sect. 2, we outline a general idea about the applicability of information granules and their refinements in building a granular classifier. Section 3 describes the classifier algorithm and its variations. In Sect. 4, we present experimental results using synthetic data sets and real data sets (Bache and Lichman 2013). We conclude the study in Sect. 5 offering some conclusions.

In the entire study, we consider N patterns (data) $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ positioned in an n -dimensional space of real numbers \mathbf{R}^n . In the classification problem, we assume that the patterns belong to d classes, $\omega_1, \omega_2, \dots, \omega_d$.

2 A general idea

As already highlighted in the previous section, the main idea of using information granules is to abstract a set of data into

a collection of sets, such that the diversity (homogeneity) of each set is sufficiently low. On the other hand, we need to keep the number of information granules reasonably low. These are two contradictory goals that can be achieved by starting with a predefined set of a few information granules. In the sequel, the goal is to refine these information granules as needed to produce more information granules with lower diversity. This refinement process is carried out by splitting the more diverse information granule into a number of less diverse, specialized information granules. In this way, a new data item can be classified to one of these information granules based on how close this item (in term of its attributes) to these information granules is.

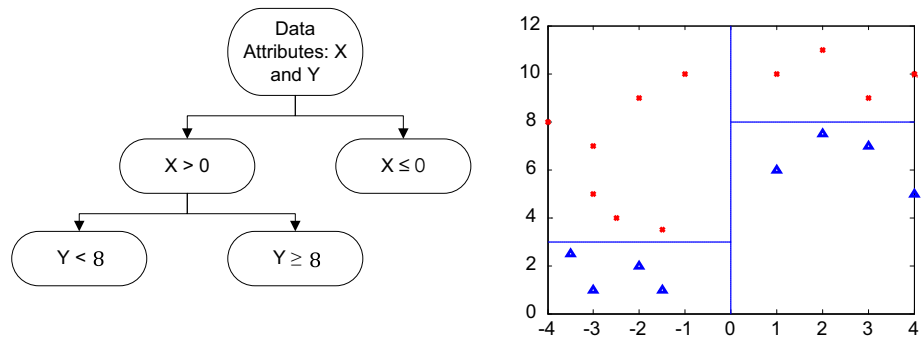
This idea is similar to the one behind decision trees (Kohavi and Quinlan 2002; Quinlan 1986) where the tree starts with a single node where every data point is a member of this node, and then the tree is refined into several nodes at the lower levels of the tree. The objective is to form successive nodes so that the nodes at the lower level tend to become more homogeneous and capture (contain) data points that can be regressed using simple models (*regression trees*) or they belong only to a single class of data points (*decision trees*) (Breiman et al. 1984; Loh and Vanichsetakul 1988; Loh and Shih 1997; Kim and Loh 2003; Loh 2002, 2009, 2011; Kim and Loh 2001; Therneau and Atkinson 2011; Chaudhuri et al. 1994; Ciampi 1991; Wang et al. 2015). This is done using some conditions imposed by the data attributes that guide the development of the tree (refer to Fig. 1). It is noticeable that the classification boundaries are piecewise linear. Furthermore, the only type of boundaries being produced by the tree result through so-called guillotine cuts (being parallel to the coordinates). Furthermore, each boundary is built on the basis of a single variable, so when traversing the tree, the boundaries are formed by selecting a suitable feature in the input space.

In contrast to decision trees, the granular classifier (Pedrycz et al. 2008) builds on a basis of information granules. Its schematic view, along with the character of the decision boundaries, is illustrated in Fig. 2.

Moreover, the boundaries among information granules (and subsequently classification boundaries) are non-linear and being formed in the entire feature space (viz. involving all input variables).

In a certain way, one may point at some similarities between the architectures of granular classifiers and radial basis function (RBF) neural networks (Broomhead and Lowe 1988). There are, however, evident conceptual and development differences. First, the RBF neural networks typically exploit Gaussian receptive fields with adjustable spreads (whose values are tuned experimentally or selected in advance). Second, there is no effect of the refinement of RBFs so that the network could grow by enhancing its accuracy.

Fig. 1 The decision tree and its refinement along with the resulting decision boundaries: 8 is the threshold of the variable Y



The underlying idea of the algorithm is as follows. Assuming that we start at the highest level of abstraction with c information granules, denoted by A_1, A_2, \dots, A_c , a successive refinement is realized by selecting the most suitable information granule based upon the diversity of its content. In this way, a refined information granule A_j is expanded to produce c more detailed (refined) information granules, denoted by $A_{j1}, A_{j2}, \dots, A_{jc}$. Once the first expansion has been completed, there are in total $2c-1$ information granules (that is $A_1, A_2, \dots, A_{j-1}, A_{j1}, A_{j2}, \dots, A_{jc}, A_{j+1} \dots A_c$), and any one of these can be a candidate for further refinements. This expansion process leads to the information granules that satisfy the condition $\sum_{i=1}^{j-1} u_{ik} + \sum_{l=1}^c u_{jlk} + \sum_{i=j+1}^c u_{ik} = 1$, where u_{ik} is the membership of the data point (pattern) x_k in the information granule i , and u_{jlk} is the membership of the data point x_k in the information granule jl . The overall idea portrayed in Fig. 3a–f illustrates the process. In Fig. 3a, we visualize a two-dimensional data set with three classes, denoted here by o, Δ , and x. In Fig. 3b, there is the highest level of abstraction where two clusters were produced using the fuzzy C-means (FCM) algorithm. If we just look at the fractions of patterns belonging to the individual classes, we find that cluster 1 exhibits a certain level of heterogeneity expressed by the mixture of patterns belonging to the individual classes [0.4 0.5 0.1], whereas cluster 2 comes with the values [0.25 0.125 0.625]. It is clear that cluster 2 is dominated by the “x” class and is thus less diverse (more homogeneous), which indicates that cluster 1 is the candidate information granule for refinement (splitting). Figure 3c shows the first refinement step, which is done for cluster 1 of Fig. 3b. Now, looking again for the fractions of patterns belonging to the classes, we find that the three clusters are characterized by information content expressed as [0.57 0.29 0.14], [0 1 0], and [0.25 0.125 0.625], respectively. It is clear that the diversity of cluster 2 is 0 (It is homogenous by being composed of patterns belonging to a single class, Δ). Again, cluster 1 is the most diverse cluster and as such it is a candidate for further refinement. This refinement is shown in Fig. 3d. Proceeding with the process, Fig. 3e, f shown are two further refinement steps.

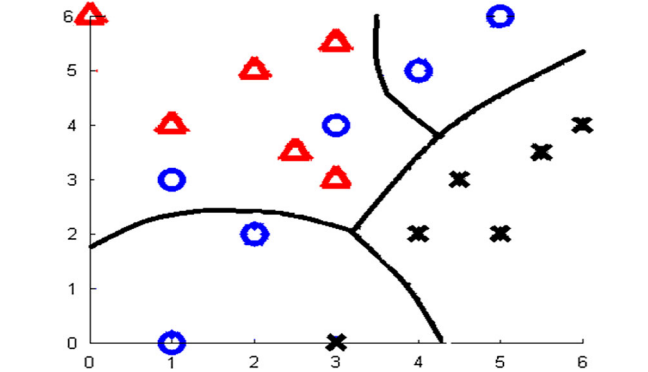


Fig. 2 The architecture of the granular classifier and its refinements completed on a basis of specialization of selected information granules

3 Algorithmic aspects of the classifier

In this section, we elaborate on the essential functional modules of the granular classifier and discuss their realization.

3.1 Construction of Information Granules and Their Information Content

The formation of information granules is realized through clustering the data into c clusters. Out of a plethora of clustering techniques, we consider here FCM (Bezdek 1981; Dunn 1973). There are several compelling reasons behind this selection. The method is broadly documented in the literature and comes with a wealth of applications. The method produces information granules that provide a comprehensive insight into the data by admitting membership grades assuming values in the [0,1] interval rather than a 0-1 quantification produced, for instance, by k -means. Not repeating the well-known material, which is well documented in the existing literature, we only briefly highlight the essence of the method and a form of the results produced by it. FCM is aimed at the minimization of a certain objective function, and its minimum is determined by running a certain iterative optimization scheme. The result of clustering of X into c clusters is provided in the form of the prototypes v_1, v_2, \dots, v_c

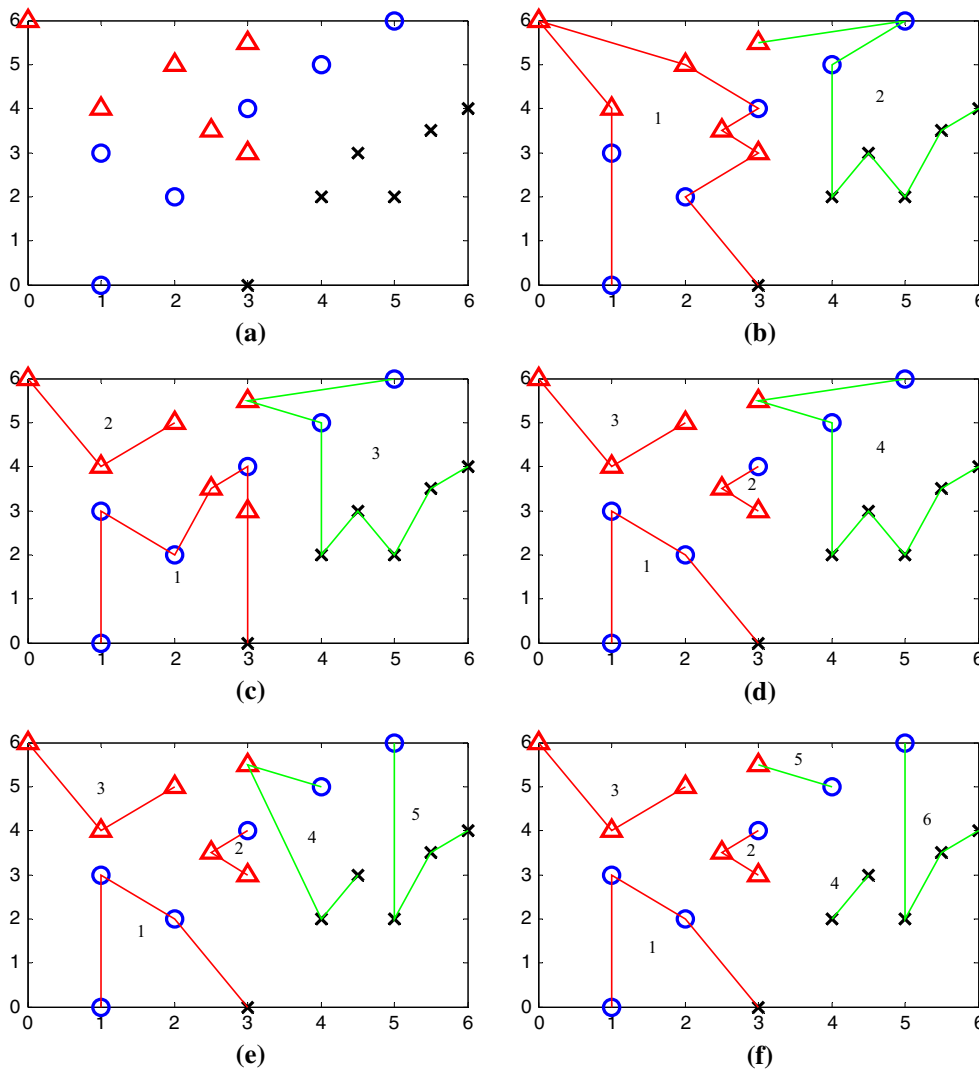


Fig. 3 Illustration of the functioning of the algorithm

and a partition matrix $U = [u_{ik}]$, $i = 1, 2, \dots, c$; $k = 1, 2, \dots, N$ describing degrees of membership of data to the individual clusters. Individual rows of the partition matrix U contain membership grades of the constructed fuzzy sets. Each information granule produced in this way, say A_1, A_2, \dots, A_c is described analytically in the following manner:

$$A_i(x) = \frac{1}{\sum_{j=1}^c \left(\frac{\|x-v_i\|}{\|x-v_j\|} \right)^{2/(m-1)}}, \tag{1}$$

where $\|.\|$ is the Euclidean distance and $m, m > 1$ is a fuzzification coefficient (Bezdek 1981). Obviously if $x = v_i$, $A_i(v_i) = 1$; Alluding to the partition matrix, we have the relationship $u_{ik} = A_i(x_k)$.

Having the revealed structure of the data X described by A_1, A_2, \dots, A_c , we can also associate with these infor-

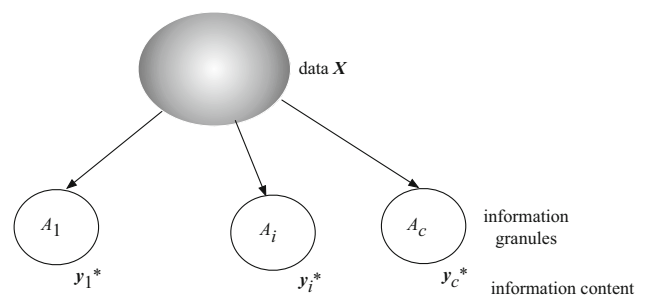


Fig. 4 A collection of information granules A_i and their information content y_i^* produced through data clustering (FCM)

mation granules the corresponding information content; see Fig. 4.

The information content implies the usefulness of the corresponding information granules in the ensuing classification

activities. In what follows, we outline several ways of quantifying this content.

We start by defining a collection of data belonging to the i -th cluster and denote this collection by X_i ,

$$X_i = \{x_k | u_{ik} = \max_{j=1,2,\dots,c} u_{jk}\} \tag{2}$$

In other words, X_i is composed of the data points that belong to the i -th cluster to the highest extent (higher than to other clusters).

In general, X_i is a mixture of data belonging to different classes and contributing to X_i itself with varying membership degrees u_{ik} . Note that we require that $c \geq d$ to achieve potentially a situation where X_i becomes homogeneous; viz., it comprises only patterns belonging to a single class.

The membership degrees of the data to the cluster and the information about class membership are the two characteristics, using which we describe information content.

Several viable alternatives are discussed below; we also include some motivation behind each of the options.

A1. We determine accumulated values of membership of the data belonging to X_i and class ω_l by computing the sum

$$Z_{il} = \sum_{k: x_k \in X_i, x_k \in \omega_l} u_{ik} \tag{3}$$

This could be seen as a certain class-driven version of a σ -count as discussed in fuzzy sets. In the sequel, we form a d -dimensional vector y_i^* coming in the form

$$y_i^* = \left[\frac{z_{i1}}{\sum_{r=1}^d Z_{ir}} \frac{z_{i2}}{\sum_{r=1}^d Z_{ir}} \cdots \frac{z_{id}}{\sum_{r=1}^d Z_{ir}} \right] \tag{4}$$

where y_i^* is a descriptor of the information content of the i -th cluster. If only one coordinate of this vector is close to 1 with others being close to 0, we say that the cluster is homogeneous. The most heterogeneous situation is encountered when all entries of y_i^* are equal to each other and close to $1/d$.

A2. This descriptor of information content is built on a basis of A_l by considering the entries of the above vector (equation 4) to be set to 0 or 1. One assigns 1 to the highest entry of y_i^* while all remaining are set up to 0. Thus we obtain a Boolean vector y_i^*

$$y_i^* = [00 \dots 010 \dots 0] \tag{5}$$

with the j_0 -th nonzero entry $j_0 = \arg \max_j z_{ij}$. In light of the formation of this information content, we can consider this description to be a less detailed (binary) version of (4), not including detailed membership grades.

A3. Here, we form y_i^* by considering counts of data belonging to cluster X_i and the corresponding classes. N_{ij} denotes

a count (number) of patterns belonging to X_i and class ω_j . We take the ratios (which, in essence, are probabilities of classes of the patterns present in the i -th cluster).

$$y_i^* = \left[\frac{N_{i1}}{\sum_{r=1}^d N_{ir}} \frac{N_{i2}}{\sum_{r=1}^d N_{ir}} \cdots \frac{N_{id}}{\sum_{r=1}^d N_{ir}} \right] \tag{6}$$

In Sect. 4, we explore all of these options through experiments of synthetic and real traces.

3.2 Splitting criterion

Once a given information granule i has been associated with the information content y_i^* , the diversity of the information granule can be quantified. We call this diversity value, the class membership content. There are several viable options to determine the value of class membership content.

B1. In this option, we consider the Euclidean distance between the information content of the information granule and the target output (class belongingness) of the data points belonging to this information granule

$$V_i = \sum_d \sum_{k=1}^{N_i} (Y_i^* - Y_k)^2, \tag{7}$$

where N_i represents the total data points belonging to information granule i . The information granule with the highest class membership content is the candidate for further refinements (splitting).

B2. Another way to model the class membership content of an information granule is to compute the entropy of the information granule information content

$$V_i = - \sum_d y_i^* \log y_i^* \tag{8}$$

3.3 Refinement process

The splitting criterion outlined above is used to select which of the c (A_1, A_2, \dots, A_c) information granules to consider as a candidate for refinement because of its too-high diversity. Assuming that because of the detected diversity, information granule A_j is the next one to refine, then in the refinement scheme, we split A_j into c information granules, say ($A_{j1}, A_{j2}, \dots, A_{jc}$) such that for any data point x_k , the following condition becomes satisfied:

$$\sum_{i=1}^{j-1} u_{ik} + \sum_{l=1}^c u_{jlk} + \sum_{i=j+1}^c u_{ik} = 1 \tag{9}$$

The membership degrees of belongingness to the jl -th sub-cluster u_{jlk} is computed using u_{jk} and the new set of

prototypes generated by applying the FCM on the candidate information granule as follows:

$$u_{jlk} = \frac{u_{jk}}{\sum_{t=1}^c \left(\frac{\|x_k - v_{jt}\|}{\|x_k - v_{jt}\|} \right)^{2/(m-1)}}, \tag{10}$$

where v_{jt} is the prototype of sub-cluster t generated from splitting the cluster j into c sub-clusters.

To clarify this process, in the following we show a numerical example from a simulation experiment. Let us consider two data-points x_q and x_l that belong to the information granule j (before any refinements), and x_q belongs to classification class 1, while x_l belongs to classification class 2. The membership values of both data-points to the information granule j were found to be: $u_{jq} = 0.5105$, and $u_{jl} = 0.8522$. When splitting the information granule j data-points to three new information granules, the membership values of x_q and x_l to these new information granules were computed using (10), but without multiplying by u_{jk} ($k = q$ or l). These membership values were found to be

$$U_q = \begin{bmatrix} 0.0024 \\ 0.0084 \\ 0.9892 \end{bmatrix}, \text{ and } U_l = \begin{bmatrix} 0.0013 \\ 0.9760 \\ 0.0227 \end{bmatrix}$$

It is clear that both of them add up to 1. Now when replacing information granule j by these three new information granules, the memberships of x_q and x_l must add up to u_{jq} and u_{jl} , respectively. To fix this, we multiply these membership values by u_{jk} in (10). Doing so, we get the following memberships for both x_q and x_l :

$$U_q = \begin{bmatrix} 0.0012 \\ 0.0043 \\ 0.5050 \end{bmatrix}, \text{ and } U_l = \begin{bmatrix} 0.0011 \\ 0.8317 \\ 0.0194 \end{bmatrix}$$

Note that this refinement process separated the two data points into two different information granules (based on the maximum value of their membership matrix), and since they belong to different classes, this reduces the diversity of the new generated information granules compared to the original information granule j . We need to make it clear here that this can happen to most of the data points of different classes assuming that they exhibit different characteristics based on their feature values.

3.4 Classification of a new pattern

Once the clusters (information granules) have been endowed with their information content, the overall architecture is used to determine class membership of a new pattern x . This process is realized in two steps:

1. Determination of activation level (membership values) of x to A_1, A_2, \dots, A_c using (1).

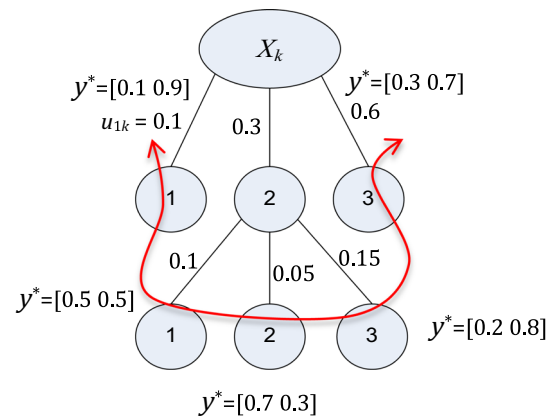


Fig. 5 Refinement of information granule present at the lower level of the tree

2. Computing the vector of class membership of the pattern x , $y = [y_1 \ y_2 \ \dots \ y_d]$, where the j -th coordinate of y comes as the following weighted sum of the information contents of the clusters; the weights are the membership values computed above. We have

$$y_j = \sum_{i=1}^c A_i(x)y_{ij}^* \tag{11}$$

$j = 1, 2, \dots, d$; $y = [y_1 \ y_2 \ \dots \ y_d]$. At the end, we select the class j_0 for which y attains its maximal value, while the vector y_i^* is computed using one of the alternatives A1–A3 as described above.

A more general aggregation mechanism is built as follows:

$$y_j = \sum_{i=1}^c A_i(x)\phi(y_{ij}^*), \tag{12}$$

where $\phi: [0,1] \rightarrow [0,1]$ is a certain non-decreasing function. Another extension could endow ϕ with some adjustable parameters.

As an illustrative example, consider the tree of information’s granules shown in Fig. 5. The degree to which the data point x_k is associated with the two classes denoted by ω_1 (1) and ω_2 (2) is computed as follows: $y = 0.1*[0.1 \ 0.9] + 0.1*[0.5 \ 0.5] + 0.05*[0.7 \ 0.3] + 0.15*[0.2 \ 0.8] + 0.6*[0.3 \ 0.7] = [0.3050 \ 0.6950]$. Therefore, x_k is classified as belonging to class 2, with a 0.695 membership degree while also exhibiting a lower level of membership (0.305) to class 1.

4 Experimental results

In this section, we present the performance of the granular classifier using synthetic data and several publicly available

data. The quality of the classifier is selected to be a classification error rate and is computed as follows:

$$\text{Error} = \frac{\sum_{k=1}^N (\tilde{Y}_k \neq Y_k)}{N}, \tag{13}$$

where \tilde{Y}_k and Y_k are the predicted class and the actual class for a data point x_k , respectively.

4.1 Synthetic data

Here we consider a two-dimensional data set of two classification classes. The two classes are separated by a continuous circular boundary, as shown in Fig. 6. The data points lying inside or on the circular boundary are considered to belong to the first class of patterns (denoted by “o”), and the data points outside the circular boundary form the second class (denoted by “x”). The data points are randomly selected in

the 2D space where each variable is defined in $[-15, 15]$, and the circular boundary is centered at the origin with a radius of 10. There are 340 data points of class “o”, and 660 data points of class “x”. We use a tenfold cross-validation scheme. The data points are randomly divided into ten groups, where in each run one of these groups is considered as the test group, and the remaining patterns are considered to be the training data.

For the purpose of illustration, we fix the values of c and m to 3 and 1.1, respectively. We first present the result of a sample run to show the performance progress as a function of the refinement process. In this sample run, we only consider options A1 and B1 to compute the values y_i^* (4) and V_i , (7), respectively. Figure 7 shows the training data and the testing data for this sample run, where the testing data represent 10 % of the overall data (tenfold cross validation).

To visualize the performance of the classifier, we display the values of the classification error as a function of the number of refinement steps (splits) for all the options of y_i^* and V_i . See Fig. 8. In this experiment, we fix the values of c and m to 3 and 1.1, respectively. This is done to illustrate the effect of the refinement process, and in the coming experiments, in the sequel, we study the effect of these two parameters (c and m) on the performance of the classifier. Figure 8 shows that although all the options produce good performance, the A1, along with the B2, leads to the best result.

To test the effect of the other parameters (c and m) on the performance, Fig. 9 shows the misclassification error (test data) for different values of m and c and a fixed number of the number of generated prototypes p that is defined as $p = c + (c - 1)N_s$, where N_s is the number of splits. We use the number of prototypes rather than the number of splits to have a fair comparison since, for a higher value of c , we

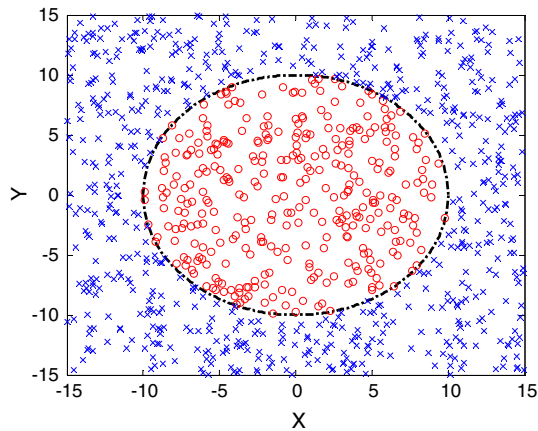


Fig. 6 2D synthetic data with two classes x and o

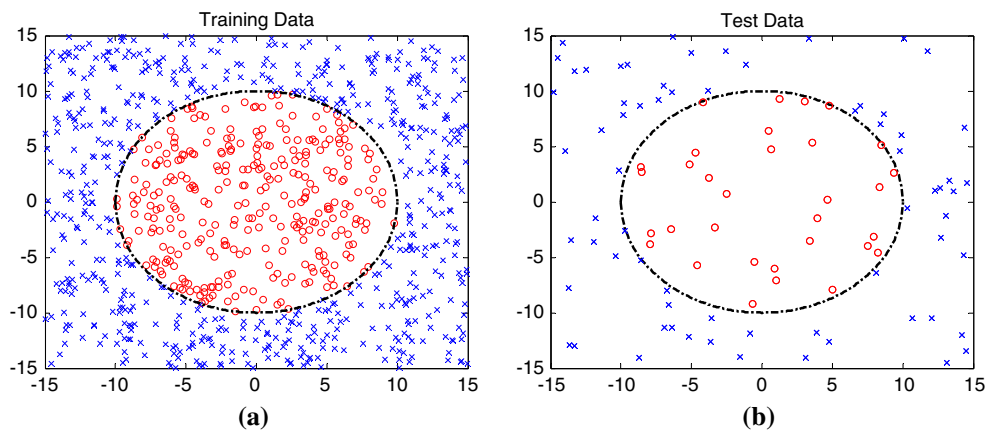


Fig. 7 Synthetic data: a training data, and b testing data

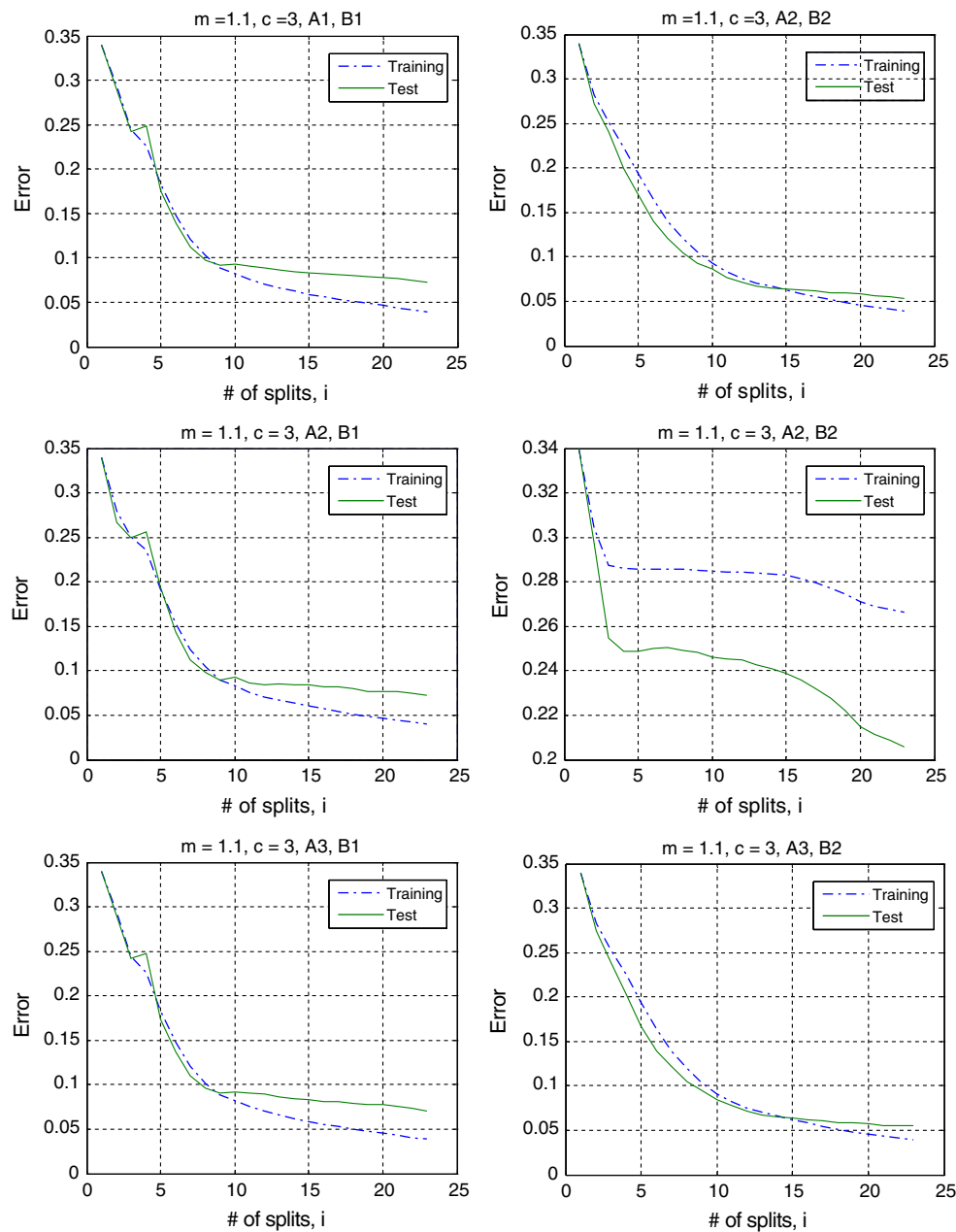


Fig. 8 Classification error rate for the synthetic data for selected combinations of values of m and c

get more prototypes at the same number of splits. We use different values for c (3, 5, 7, and 9) and different values for m (1.1, 1.3, 1.5, 1.7, and 2). We do the refinement to generate up to 49 prototypes. This value is selected so that the corresponding number of splits, N_s , has no fractions for the values of c . Accordingly, the number of splits for the different

values of c is 23, 11, 7, and 5, respectively. In general, a value of m less than 2.0 (between 1.5 and 1.7) gives better performance than using higher values of the fuzzification coefficient. Moreover, using a low value for c (between 3 and 5) gives better performance than using high values. This is logical, since we have a limited number of refinements and

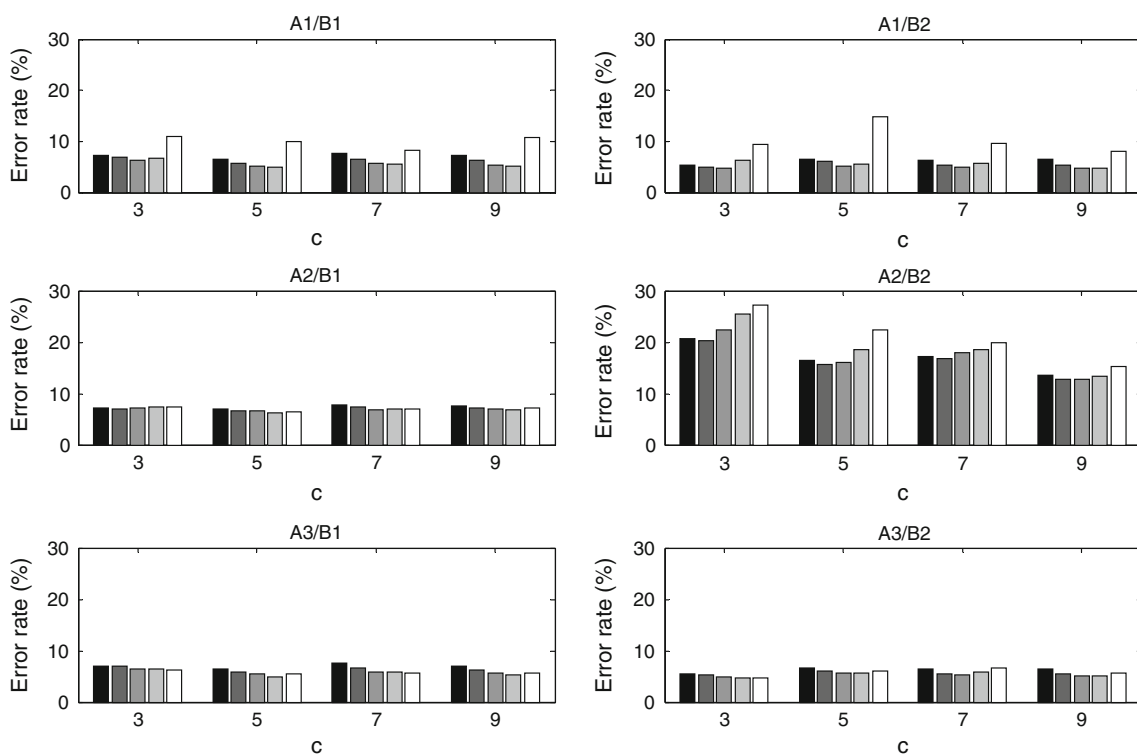


Fig. 9 Classification error rate as a function of m and c for the synthetic data set (test data)

decreasing the c value gives a chance for more information granules to be less diverse. The case A2/B2 is different from the other cases since this case is like a random case where the cluster to refine is randomly selected. This is because of the entropy computation for all values of y^* is the same, since the content of the vector y^* are only zeroes and ones, and, in this case, all information granules are seen as if they had the same diversity.

4.2 Machine learning data

In this section, we demonstrate the applicability of the scheme in classification using the machine learning data [Bache and Lichman \(2013\)](#). We use eight data sets as reported in [Table 1](#). These data sets represent diverse data sets in terms of the number of data, the number of attributes (features), and the number of classes. In the first experiment, we show the effect of m and c on the performance of the classifier in the

Table 1 Selected machine learning data sets

Data set	# Examples	# Classes	# Attributes
Ionosphere	351	2	34
Liver disorder	345	2	7
Pima diabetes	768	2	8
Segment	2310	7	19
Tic-Tac-To	958	2	9
Vehicle	846	4	18
Vowel	990	11	10

same way as we did for the synthetic data. In [Fig. 10a–g](#), we show the classification error (for the testing data) for different values of m and c when fixing the number of the prototypes (information granules) as we did before. The refinement is continued up to the point where 49 prototypes have been generated. From these plots, several conclusions are drawn.

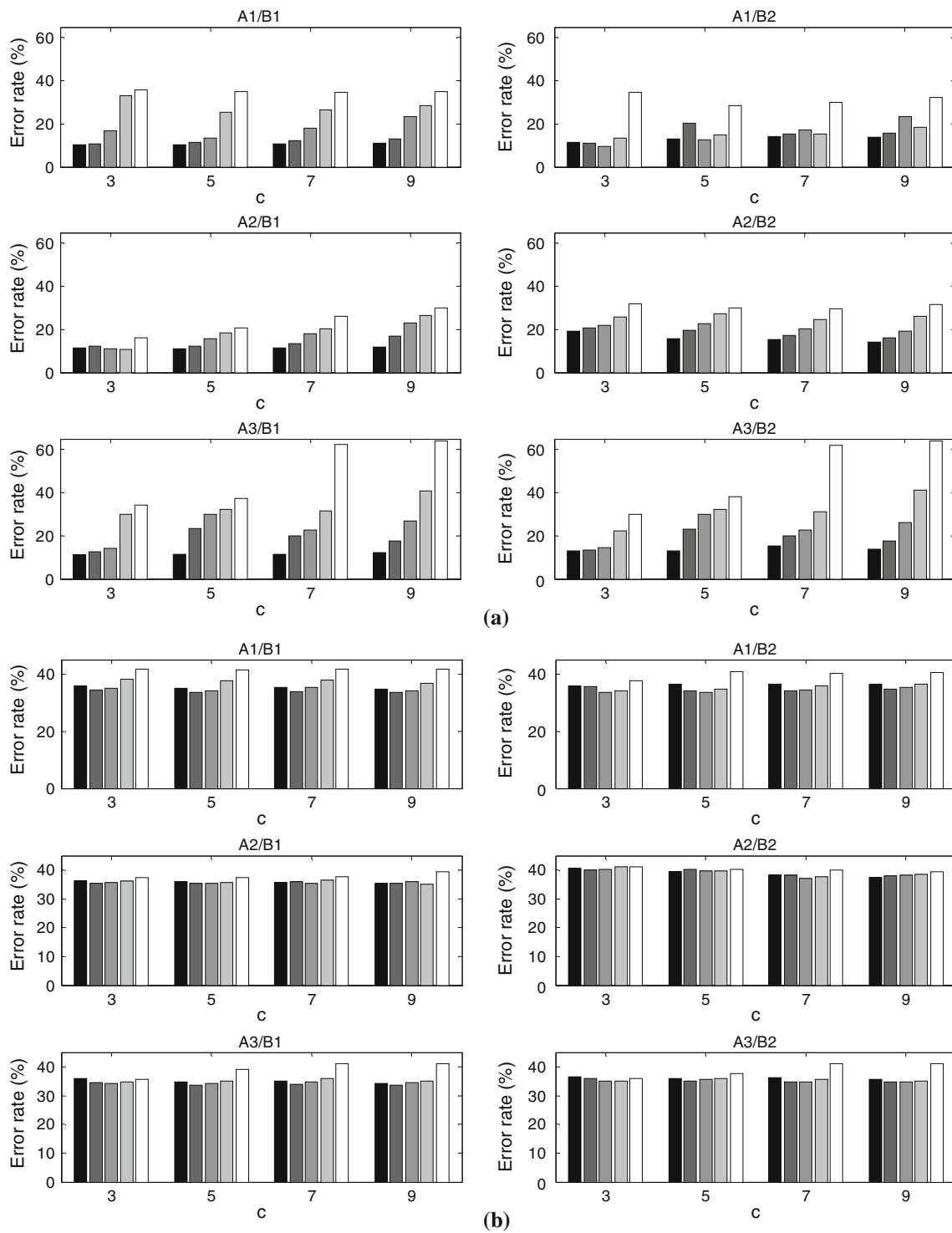


Fig. 10 **a** Classification error rate as a function of m and c for the Ionosphere data set (testing data). **b** Classification error rate as a function of m and c for the liver disorder data set (testing data). **c** Classification error rate as a function of m and c for the pima diabetes data set (testing data). **d** Classification error rate as a function of m and c for the seg-

ment data set (testing data). **e** Classification error rate as a function of m and c for the tic-tac-to data set (testing data). **f** Classification error rate as a function of m and c for the vehicle data set (testing data). **g** Classification error rate as a function of m and c for the vowel data set (testing data)

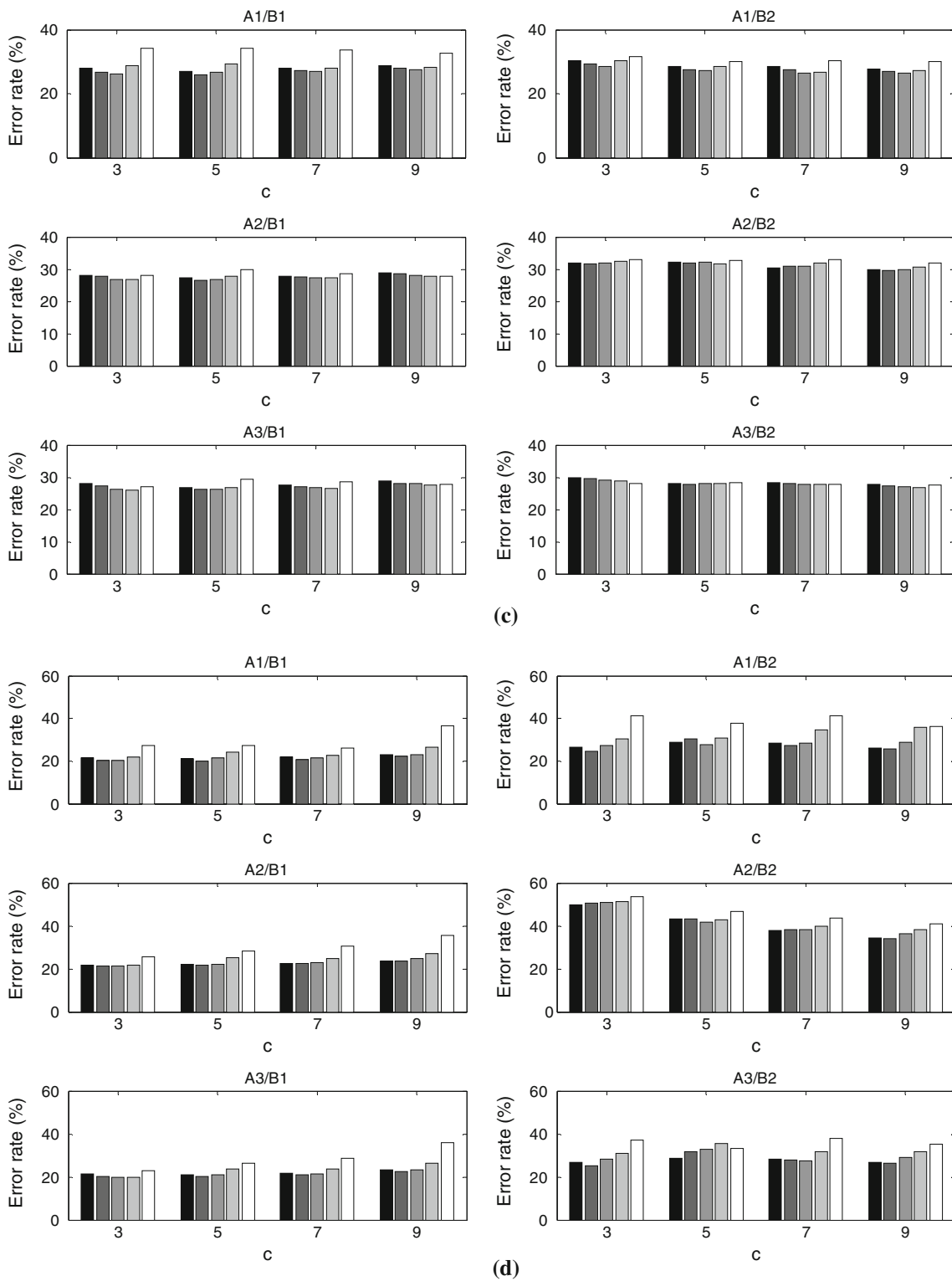


Fig. 10 continued

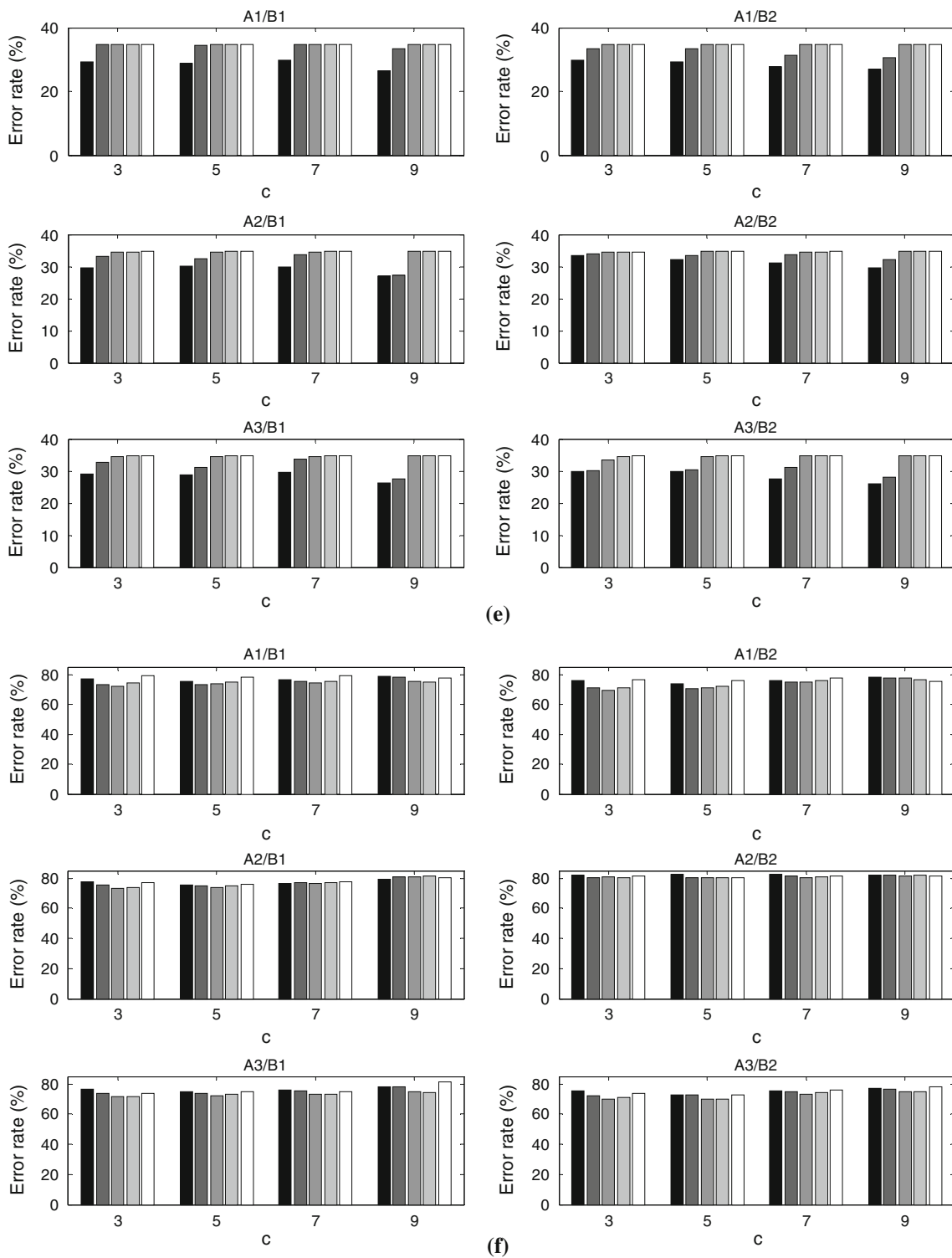


Fig. 10 continued

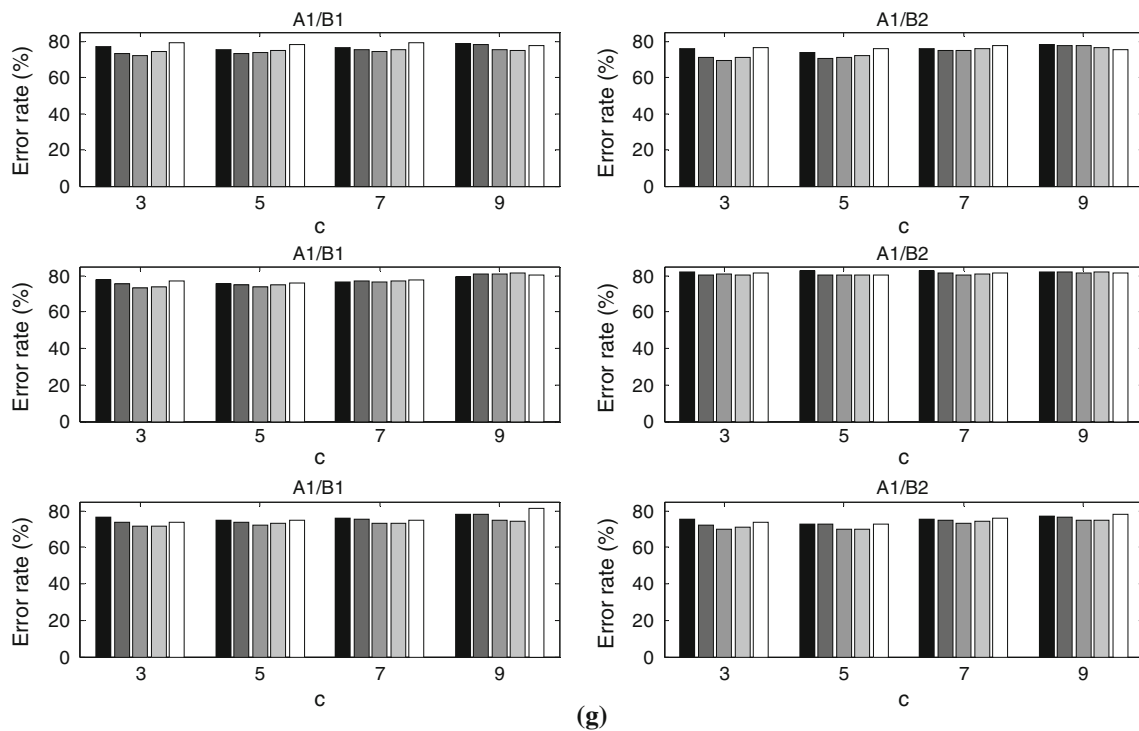


Fig. 10 continued

We can see that, in most cases, a value of m less than 2 gives better performance than when using the fuzzification index assuming higher values. Moreover, using a low value of c (ranging between 3 and 5) gives better performance than using high values of c . This is not true for the Tic-Tac-To/Vehicle traces (Figs. 10e/10f, 11e/11f), where the best performance is achieved for c assuming values in the range from 7 to 9. Moreover, the A1 option (Eq. 4) seems to be the best criteria for computing the y^* value, and the B2 option (Eq. 8) seems to be better than the B1 option (Eq. 7) for computing the value of V_i . In the series of plots, Fig. 11a–g, we display classification error rate regarded as a function of the number of splits for all the combinations of the values m and c that give the best performance (according to Fig. 10a–g).

5 Conclusions

The proposed granular classifiers exploit a fundamental concept of information granules, which are crucial to build-

ing classification mappings that are both nonlinear (and as such become capable of coping with classification problems that are not linearly separable) and interpretable (owing to the fact that information granules associated with some underlying semantics). The stepwise refinement of information granules with regard to a successive improvement of their information content becomes crucial to the enhancement of the quality of the resulting classifier and helps establish a sound tradeoff between accuracy and the conciseness (compactness) of the resulting construct.

There are several interesting and promising directions for further studies. First, information granules can be formalized in many different ways (as studied in granular computing (Pedrycz 2005; Bargiela and Pedrycz 2003; Pedrycz 2001; Lin 2003)) using sets, fuzzy sets, rough sets, and the like in comparison to fuzzy sets used in this study. Second, more alternatives to aggregate information granules could be sought while making the detailed mappings adjustable by endowing them with parameters whose values can be tuned during the learning process.

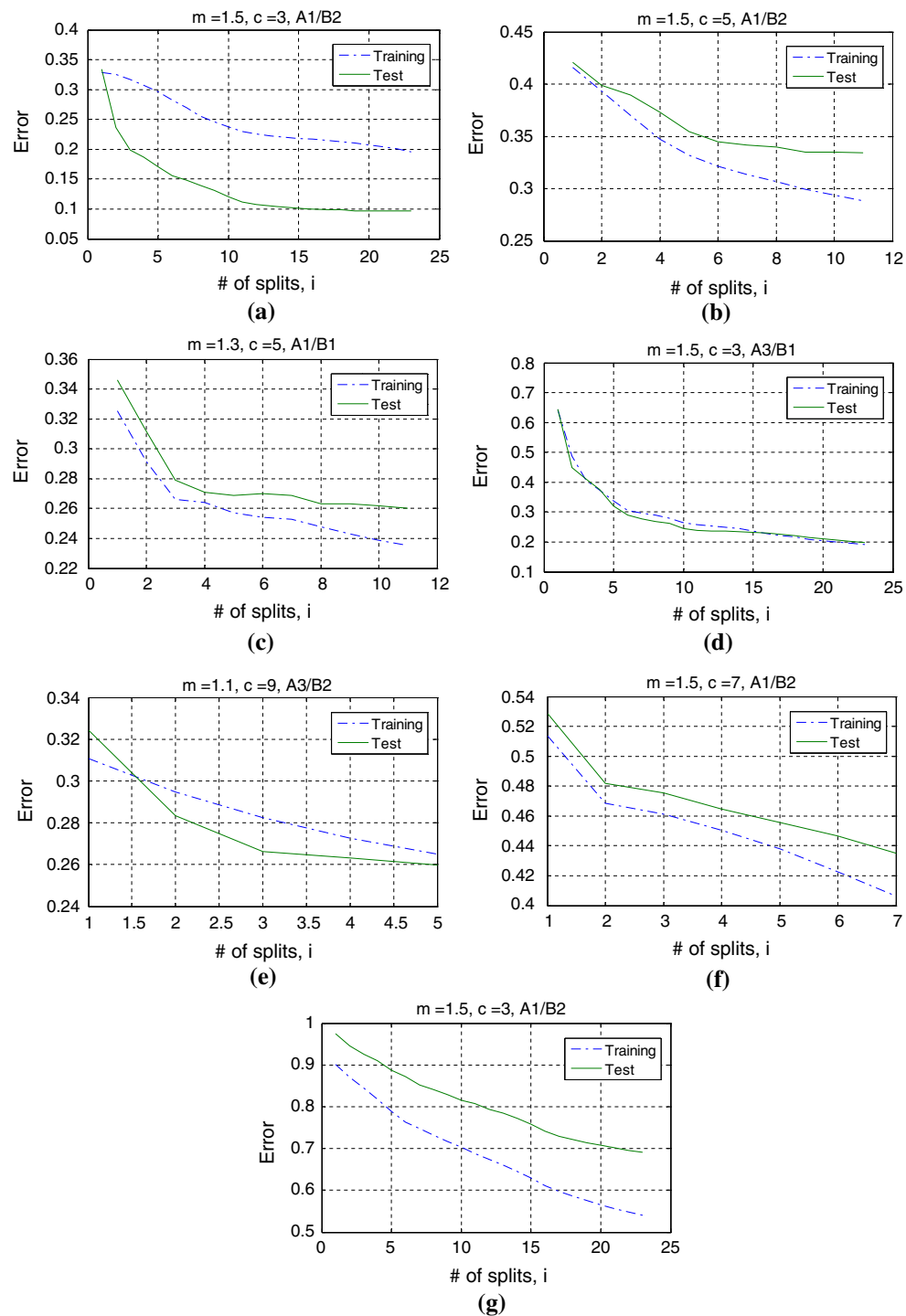


Fig. 11 **a** Classification error rate as a function of the number of splits for the best configuration for Ionosphere data set (testing data). **b** Classification error rate as a function of the number of splits for the best configuration for Liver data set (testing data). **c** Classification error rate as a function of the number of splits for the best configuration for Pima diabetes data set (testing data). **d** Classification error rate as a function of the number of splits for the best configuration for Segment data set

(testing data). **e** Classification error rate as a function of the number of splits for the best configuration for Tic-Tac-toe data set (testing data). **f** Classification error rate as a function of the number of splits for the best configuration for Vehicle data set (testing data). **g** Classification error rate as a function of the number of splits for the best configuration for Vowel data set (testing data)

Acknowledgements This article was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under Grant No. (135-804-D1435). The authors, therefore, gratefully acknowledge the DSR technical and financial support.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

- Al-Hmouz R, Pedrycz W, Balamash A, Morfeq A (2014) From data to granular data and granular classifiers. In: 2014 IEEE international conference on fuzzy systems (FUZZ-IEEE), pp 432–438
- Al-Hmouz R, Pedrycz W, Balamash A (2015) Description and prediction of time series: a general framework of granular computing. *Expert Syst Appl* 42:4830–4839
- Apolloni B, Pedrycz W, Bassis S, Malchiodi D (2008) The puzzle of granular computing. Springer, Heidelberg
- Bache K, Lichman M (2013) UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml>. Accessed June 2015
- Balamash A, Pedrycz W, Al-Hmouz R, Morfeq A (2015) An expansion of fuzzy information granules through successive refinements of their information content and their use to system modeling. *Expert Syst Appl* 42:2985–2997
- Bargiela A, Pedrycz W (2003) *Granular computing: an introduction*. Springer Science & Business Media, Berlin
- Bezdek JC (1981) *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, Norwell
- Breiman L, Friedman J, Stone CJ, Olshen RA (1984) *Classification and regression trees*. CRC Press, Boca Raton
- Broomhead DS, Lowe D (1988) Radial basis functions, multi-variable functional interpolation and adaptive networks. DTIC Document 1988
- Chaudhuri P, Huang M-C, Loh W-Y, Yao R (1994) Piecewise-polynomial regression trees. *Stat Sinica* 4:143–167
- Ciampi A (1991) Generalized regression trees. *Comput Stat Data Anal* 12:57–78
- Dunn JC (1973) A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J Cybern* 3(3):32–57
- Kim H, Loh W-Y (2001) Classification trees with unbiased multiway splits. *J Am Stat Assoc* 96:589–604
- Kim H, Loh W-Y (2003) Classification trees with bivariate linear discriminant node models. *J Comput Gr Stat* 12:512–530
- Kohavi R, Quinlan JR (2002) Data mining tasks and methods: classification: decision-tree discovery. In: *Handbook of data mining and knowledge discovery*. Oxford University Press, Inc., pp 267–276
- Lin TY (2003) Granular computing. In: *Rough sets, fuzzy sets, data mining, and granular computing*. Springer, pp 16–24
- Loh W-Y (2002) Regression trees with unbiased variable selection and interaction detection. *Stat Sinica* 12:361–386
- Loh W-Y (2009) Improving the precision of classification trees. *Ann Appl Stat* 3(4):1710–1737
- Loh W-Y (2011) Classification and regression trees. *Wiley Interdiscip Rev: Data Min Knowl Discov* 1:14–23
- Loh W-Y, Shih Y-S (1997) Split selection methods for classification trees. *Stat sinica* 7:815–840
- Loh W-Y, Vanichsetakul N (1988) Tree-structured classification via generalized discriminant analysis. *J Am Stat Assoc* 83:715–725
- Pedrycz W (2001) *Granular computing: an emerging paradigm*, vol 70. Springer Science & Business Media, Berlin
- Pedrycz W (2005) *Knowledge-based clustering: from data to information granules*. Wiley, New York
- Pedrycz W (2013) *Granular computing: analysis and design of intelligent systems*. Wiley, Hoboken, New Jersey
- Pedrycz W, Park B-J, Oh S-K (2008) The design of granular classifiers: a study in the synergy of interval calculus and fuzzy sets in pattern recognition. *Pattern Recognit* 41:3720–3735
- Quinlan JR (1986) Induction of decision trees. *Mach Learn* 1:81–106
- Therneau TM, Atkinson EJ (2011) An introduction to recursive partitioning using the RPART routines. Technical report 61. <http://www.mayo.edu/hsr/techrpt/61.pdf1997>
- Wang X, Liu X, Pedrycz W, Zhang L (2015) Fuzzy rule based decision trees. *Pattern Recognit* 48:50–59