Malaysian Journal of Computer Science, Vol. 18 No. 1, June 2005, pp. 31-39

A FUZZY VALIDITY-GUIDED PROCEDURE FOR CLUSTER DETECTION

Mohamed Benrabh, Abdelaziz Bouroumi and Abdellatif Hamdoun
Laboratoire de Traitement de l'Information
Faculté des Sciences Ben M'sik
Université Hassan II-Mohammedia
B.P. 7955 Sidi Othmane
Casablanca, Morocco
e-mail: benrabh@yahoo.fr
bouroumi@fsr.ac.am
alhamdoun@yahoo.fr

ABSTRACT

In this paper, we present a new procedure for detecting clusters within unlabelled data sets of the form $X = \{x_1, x_2, ..., x_n\} \subset \Re^p$. *This procedure quickly explores the elements of* X *with the main goal of discovering the clusters they form. It provides, in addition to the number of clusters, an initial prototype of each detected cluster. For this, the only assumptions made are that (1) the two least similar elements of* X *belong necessarily to two different clusters, and (2) each element possesses a level of similarity with its nearest prototype greater than a certain threshold. This threshold can be either user defined or automatically determined by the algorithm using a validation process. The effectiveness of this method is demonstrated on both synthetic and real test data sets.*

Keywords: ***Unsupervised learning, Fuzzy clustering, Cluster validity, Pattern recognition***

1.0 INTRODUCTION

Clustering is a common problem in many application domains including industry, business, telecommunications, medicine, and social sciences. Depending on the context, this problem can be found in the literature under different appellations, such as numerical taxonomy in biology and ecology, typology in social sciences, partition in graph theory, and cluster analysis or unsupervised learning in pattern recognition [1]. Formally, a clustering problem can be posed in terms of finding the best partition of a data set $X = \{x_1, x_2, ..., x_n\} \subset \Re^p$, called learning base, into c clusters or homogenous groups such that the objects of each cluster should be more similar to each others than are objects belonging to different clusters.

It is a NP-hard optimisation problem for which optimal solutions can only be found in exponential time, which is still beyond the power of modern computers [2]. In spite of this, and due to its importance in image analysis, vector quantisation, data mining, among other fields, this problem has been the subject of intensive research for many years. As a result, many sub-optimal clustering methods can be found in the literature [3-5]; most of them requiring the number of clusters as an input, although in real world applications this information is seldom available.

The purpose of this article is to introduce a new technique for automatic detection of the number of clusters. This technique is mainly based on a quick exploration of the n elements of X in order to discover the number of clusters they form. For this, a similarity measure and an associated threshold are needed. This operation provides, in addition to the number of clusters c , an initial representative or prototype of each detected cluster. The c resulting prototypes are then optimised using the well-known Fuzzy C-means (FCM) algorithm [5]. The threshold parameter can be interpreted as the minimum of similarity that each object should have with its nearest prototype. It can be either fixed by the user or automatically varied between two limits whose values are derived from the data at hand. In the latter case, more than one candidate solution may be obtained. To select the best one among these solutions, a validity process is finally performed based on a quality measure. A more formal description of this method is given in Section 3 whilst Section 2 is dedicated to a succinct presentation of unsupervised fuzzy learning and the FCM algorithm. Experimental results are discussed in Section 4. Finally, Section 5 contains our conclusion and provides some ideas for further research.

2.0 UNSUPERVISED FUZZY LEARNING

Unsupervised fuzzy learning is a generalisation of hard clustering that presents the advantage of dealing with overlapping clusters or classes [5]. It is mainly characterised by the fact that instead of assigning each datum to a unique class, it assigns to the datum a membership degree to each class [6], which is very useful in real-world applications where boundaries among clusters are usually not well-defined [7]. More formally, a partition of X into c fuzzy clusters can be defined with a $(c \times n)$ constrained membership matrix $U = [u_{ik}]$, $1 \le i \le c$, $1 \le k \le n$, where $u_{ik} \in [0,1]$ denotes the degree to which the $k\,th$ object belongs to the $i\,th$ class, and

$$\sum_{i=1}^{c} u_{ik} = 1; \;\; 1 \le k \le n \tag{1a}$$

$$0 < \sum_{k=1}^{n} u_{ik} < n; \;\; 1 \le i \le c \tag{1b}$$

The first constraint ensures that the membership of each object is distributed over all the c clusters whilst the second constraint guaranties that no cluster is empty or totally equal to X.

Given the matrix U we can derive a prototype for each class i using the fuzzy mean vector

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m}; \;\; 1 \le i \le c \tag{2}$$

where $m > 1$ is a weighting exponent whose role is explained in the next section. Inversely, given the matrix of prototypes, $V = \{v_1, v_2, ..., v_c\} \subset \Re^p$, we can easily determinate the membership degree of each object to each class according, for example, to a relation of the form

$$u_{ik} = \frac{\|x_k - v_i\|^{-q}}{\sum_{j=1}^{c} \|x_k - v_j\|^{-q}}; \;\; 1 \le k \le n; \;\; 1 \le i \le c \tag{3}$$

where $\|x_k - v_i\|$ denotes the distance between the $k\,th$ object vector and the $i\,th$ prototype, $q > 0$ is a weighting exponent, and the sum in the denominator guaranties the constraints of (1a) and (1b).

Thus, by iteratively recalculating U terms using V ones and vice versa, one can hope to converge to an optimal pair (U^*, V^*) that optimises some objective criterion; and the FCM algorithm described below is one of the most popular methods used to achieve this optimisation.

2.1 Fuzzy C-means Algorithm (FCM)

FCM is an iterative procedure conceived to minimise the objective function

$$J_m(U, V; X) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \|x_k - v_i\|^2 \tag{4}$$

which can be interpreted as a fuzzy measure of the total error incurred in representing the n samples $x_1, x_2, ..., x_n$ by the c prototypes $v_1, v_2, ..., v_c$. It is a non-convex function that possesses many local minima, corresponding each to an approximate solution, i.e. to a sub-optimal partition of X [8]. In 1981, Bezdek proved that by alternately recalculating U and V, using Eqs (2) and (3), the convergence to a local minimum of J_m is guaranteed by the constraint

$$q = \frac{2}{m-1} \tag{5}$$

where the parameter $m > 1$ serves to control the fuzziness degree of the final partition [5]. In effect, when $m \to 1$, this partition tends to be a crisp one, i.e. $u_{ik} \to 1$ or $u_{ik} \to 0 \ \forall i,k$. Inversely, when $m \to \infty$, $u_{ik} \to 1/c \ \forall i,k$; and the final partition tends to be the fuzziest one. In this work, we have chosen $m = 2$, which is by far the most commonly used value in practice [5, 9].

Thus, starting from an initial matrix of prototypes V_0, whose elements can be randomly or otherwise chosen, and using Eqs. (2), (3) and (5), FCM iteratively recalculate U_t and V_t, where t denotes the iteration index, until the c prototypes stabilise, i.e. until the difference

$$\left\| V_t - V_{t-1} \right\| = \underbrace{\max}_{1 \le k \le p} \left(\underbrace{\max}_{1 \le i \le c} \left(\left| v_{ik,t} - v_{ik,t-1} \right| \right) \right), \tag{6}$$

becomes smaller than a tolerable threshold ε, or until a maximum number of iterations $t_{\max}$ is reached.

3.0 THE CLUSTER DETECTION PROCEDURE

Assuming that the learning base X contains at least two different classes (otherwise there is no need to partition it) and given a measure of inter-points similarities, say sim, and an associated threshold, th, which can be interpreted as the minimum of similarity that each element should have with the prototype of its class, we start by seeking the pair (x_i, x_j) of the least similar elements of X. Then, considering that x_i and x_j should necessarily belong to two distinct classes, we create two first clusters ($c = 2$) initialising their prototypes v_1 and v_2 with x_i and x_j, respectively. The $n-2$ remaining objects are then successively explored and for each of these objects, say x_k, we calculate the similarity to the prototype of each previously detected class, i.e. the c classes detected before analyzing x_k ($c \ge 2$). Then, depending on the maximum of these similarities, two different cases are envisaged. The first case corresponds to

$$\max_{1 \le i \le c} \left\{ sim(x_k, v_i) \right\} < th \tag{7}$$

In this case, we consider that x_k is not recognisable. Consequently, a new class is created and its prototype initialised to x_k. In other words, as x_k is not sufficiently similar to any prototype of the previously detected clusters, it is supposed to come from a not yet discovered cluster and is, therefore, utilised to initialise the prototype of a new cluster. Thus, before analysing the next element, if any, we put $c = c + 1$ and $v_c = x_k$. In the second case, which corresponds to

$$\max_{1 \le i \le c} \left\{ sim(x_k, v_i) \right\} \ge th, \tag{8}$$

we do not need to create a new cluster because x_k can acceptably be considered as coming from an already detected cluster, probably the one whose prototype is the most similar to x_k. However, instead of completely assigning x_k to a unique cluster, we make a softer decision and partially assign it to all the previously detected classes, using the similarities $sim(x_k, v_i)$, $1 \le i \le c$, as membership degrees. As a result, the prototype of each class i becomes

$$v_i(k) = \frac{1}{n_i(k)} \sum_{j=1}^{k} sim(x_j, v_i) \times x_j; \quad 1 \le i \le c \ \text{ and } \ c \ge 2 \tag{9}$$

where $n_i(k) = \sum_{j=1}^{k} sim(x_j, v_i); \ 1 \le i \le c; \ k \le n$ denotes the fuzzy cardinality of the $i\,th$ class after processing x_k.

Rewriting the sum in Eq. (9) under the form $\sum_{j=1}^{k-1} sim(x_j, v_i) \times x_j + sim(x_k, v_i) \times x_k$, and remarking that $\sum_{j=1}^{k-1} sim(x_j, v_i) \times x_j = v_i(k-1) \times n_i(k-1)$, with $n_i(k-1) = n_i(k) - sim(x_k, v_i)$, the learning rule expressed by Eq. (9) can be reformulated as follows:

$$v_i(k) = v_i(k-1) + \frac{sim(x_k, v_i)}{n_i(k)}[x_k - v_i(k-1)]; \quad 1 \le i \le c; \quad c \ge 2 \tag{10}$$

Eq. (10) means that existing classes compete for each new object x_k whose maximum of similarity exceeds th. However, instead of updating the prototype of a single class, the winner, we update all the c prototypes according to their respective similarities with x_k. This learning scheme can be viewed as a fuzzy generalisation of the winner takes all (WTA) rule used by some vector quantisation (VQ) methods [10]. It offers a better way to exploit the global information about the geometrical structure of X carried by each new vector object x_k. Indeed, this information is not limited to the similarity of x_k with a unique prototype, but distributed over its similarities with all the c prototypes.

3.1 Similarity Measure and Similarity Threshold

In the previous section, we have explained the principle of cluster detection without discussing two important questions: (1) how can we measure the inter-points similarities? And (2) how can we choose the threshold th ? It is clear that in the absence of any prior information about the structure of X, it is quite difficult to provide general answers to these questions, i.e. answers that would be acceptable in different situations and for different data structures. In practice, however, choosing a particular measure can sometimes be guided by the shape of expected classes. It is well known, for example, that measures based on Euclidean distance favour the creation of hyper-spherical and equally sized classes. The same measures may lead to mediocre results in the case of non-spherical or unequally sized classes. In this work, the following empirical relation has been used as a default measure

$$sim(x_i, x_k) = \sqrt{1 - \frac{\|x_i - x_k\|_A^2}{p}}; \quad \forall x_i, x_k \in \Re^p \tag{11}$$

where A is the $p \times p$ matrix defined by

$$A_{jt} = \begin{cases} (r_j)^{-2}; & j = t \\ 0; & \text{otherwise} \end{cases} \tag{12}$$

and $r_j = \max_{1 \le i \le n}(x_{ij}) - \min_{1 \le i \le n}(x_{ij}); \quad 1 \le j \le p$.

The choice of this measure, which can be replaced by any other one that the user may prefer, is justified by the following properties [11]: (i) $sim(x_i, x_k) \in [0,1]; \ \forall x_i, x_k \in \Re^p$, (ii) $sim(x_i, x_k) = 1$ for $x_i = x_k$, and (iii) $sim(x_i, x_k) \to 0$ when $|x_{ij} - x_{kj}| \to r_j; \ \forall 1 \le j \le p$. More importantly, Eq. (11) can also be used as a membership degree measure, i.e. $sim(x_k, v_i) = u_{ik}$, which is an essential property for the learning rule. As to the threshold th, it can theoretically vary between the limits

$$th_{\min} = \min_{i \ne k}\{sim(x_i, x_k)\} \tag{13a}$$

and

$$th_{\max} = \max_{i \ne k}\{sim(x_i, x_k)\} \tag{13b}$$

The first limit corresponds to $c = 2$ because the condition of creating a new class (Eq. (7)) will never be verified for this special value of th. The second limit corresponds to $c = n$ because in this case, Eq. (7) will always be verified, which leads to a partition where each object forms its own singleton. Thus, by varying th between these limits using a certain step Δth, different sets of c prototypes can be detected. Note, however, that this technique is intrinsically different from repeating a procedure which requires c as an input for different c values. Furthermore, in our case nothing guaranties that a partition can be obtained for any value of c.

3.2 The Validation Problem

The above description shows that, unless a particular value of th can be confidently chosen, using different th values can lead to more than one candidate solution. To select (or validate) the best one among these solutions, we used the Xie and Beni validity index [12] defined by

$$XB(U,V;X) = \frac{1}{n}\frac{J_m(U,V;X)}{\underbrace{\min}_{i\neq r}\left\{\|v_i - v_r\|^2\right\}} \tag{14}$$

which is one of the most reliable criteria available in the literature [13, 14]. According to Eq. (14), which can be interpreted as a measure of the ratio "compactness/separation" of the fuzzy partition defined by the pair (U,V), the best solution is the one that minimises XB.

3.3 Pseudocode

The pseudo-code of the entire procedure, as we have implemented it in C++ programming language, is given below where the default value of each parameter is indicated between braces.

Given an unlabelled data set $X = \{x_1, x_2, ..., x_n\} \subset \Re^p$

1. choose a similarity measure, sim [Eq. (11)]
2. find the least similar objects, x_i and x_j
3. swap(x_1, x_i); swap(x_2, x_j) // to avoid unnecessary tests
4. choose: $th_{\min}$ [Eq. (13a)], $th_{\max}$ [Eq. (13b)], Δth [10%], $t_{\max}$ [500], ε [0.00001], m [2]
5. for each th do { put $c = 2$; $v_1 = x_1$; $v_2 = x_2$;
   - for $i = 3$ to n do
     if ($\max_{1\le j\le c}(sim(x_i, v_j) < th$) do { $c = c + 1$; $v_c = x_i$; }
     else update v_j using Eq. (13) for $j = 1,2,...,c$
   - initialise: $V_0 = (v_1, v_2, ..., v_c)$, $t = 0$ (number of iterations)
   - repeat { $t++$;
     calculate U_t using V_{t-1} and Eqs. (3) and (5); calculate V_t using U_t and Eq. (2)
     } until ($\|V_t - V_{t-1}\| < \varepsilon$ or $t = t_{\max}$)
   - Use $U^* = U$ and $V^* = V$ to calculate validity criterion $XB(U,V;X)$

   }
6. return the appropriate number of clusters, c^*, and their prototypes $V^* = (V_1^*, V_2^*, ..., V_{c^*}^*)$.

4.0 EXPERIMENTAL RESULTS

To illustrate the usefulness of the proposed algorithm, we present here some example results, obtained from a collection of five test data sets we called X1, X2, X3, Iris, and Iris23. As depicted in Figs. 1-3, X1, X2, and X3 are visibly organised into two compact and well separated clusters for X1, two less compact and slightly overlapping classes for X2, and three well separated, but not very compact classes for X3. Despite their simplicity, such two-dimensional data sets can be pedagogically important due to the possibilty of visualisation they offer.

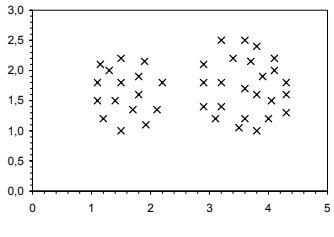

Fig. 1: Test data set X1

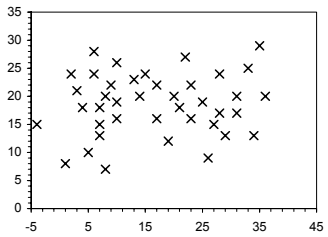

Fig. 2: Test data set X2

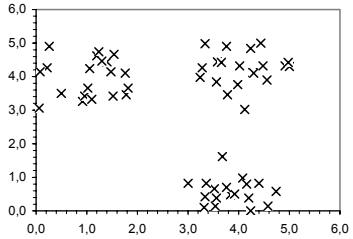

Fig. 3: Test data set X3

As a four-dimensional example, we considered Iris data, which consists of 150 real observations originated from three different varieties of flowers (Setosa, Versicolor, and Virginica), each with 50 points [15]. This is a very popular example, widely used for testing clustering algorithms; and one of its main characteristics is that the first class is well separated from the other two, which exhibit a substantial overlap. Consequently, one can argue for both $c = 2$ or $c = 3$ for this particular example. Iris23 is a subset of Iris, containing only the two overlapping clusters. This example is interesting in the sense that it permits to test the ability of clustering algorithms to separate well overlapping classes.

Tables 1 and 2 summarise the results obtained for the five examples. These results were obtained using the default value of each parameter except for Δth, whose value was fixed to 1% in order to increase the number of solutions. For each data set, the th column of these tables shows the smallest values of th that led to a number of clusters comprising between 2 and 10. We can remark that some values of c are skipped for some data sets. This is the case, for example, for X2 and $c \in \{3,6,9\}$. To verify whether an optimal result can be found for these c values in this particular example, we performed a fine tuning of th, using $\Delta th = 0.1\%$, between the limits of 62% and 63% for $c = 3$, 68% and 69% for $c = 6$, and between 74% and 75% for $c = 9$. The results of this operation are placed after those related to $c = 10$. The validity indices of these three candidate solutions (column XB) show clearly that they should be discarded. The same thing can be said for X3 and $c = 7$ or $c = 9$. Moreover, no partition with 7 classes has been detected for X3, even by varying th with a step of 0.00001% between the limits of 83% and 84%. This result signifies that, within X3, there is no structure where the elements of each class present a minimum of similarity with the prototype of that class comprising between 83% and 84%. Of course, other algorithms, including FCM, may partition any data set into any specified number of clusters regardless of whether such a partition makes sense for the given data set or not.

As to t columns of Tables 1 and 2, they show the number of iterations required for FCM to optimise each matrix of learned prototypes. We can see that the limit $t_{\max} = 500$ has never been reached. The validity indices are shown on XB columns, where optimal values are displayed in bold. We can see that for all examples except Iris, the best solution corresponds always to the actual number of clusters.

Table 1: Summary of the results obtained for the three artificial data sets X1, X2, and X3

X1				**X2**				**X3**			
th(%)	**c**	**t**	**XB**	**th(%)**	**c**	**t**	**XB**	**th(%)**	**c**	**t**	**XB**
15.45	**2**	12	**0.069**	10	**2**	17	**0.118**	10.00	2	27	0.131
42.45	3	112	0.280	63	4	92	0.202	39.00	**3**	13	**0.049**
51.45	4	80	0.298	65	5	134	0.154	55.00	4	29	0.631
66.45	5	89	0.214	69	7	58	0.196	78.00	5	28	0.459
73.45	6	177	0.164	70	8	115	0.276	81.00	6	33	0.298
74.45	7	175	0.148	75	10	210	0.199	84.00	8	65	0.192
77.45	8	147	0.132	62.2	3	152	0.226	86.00	10	61	0.220
79.45	9	47	0.132	68.3	6	66	0.176	?	7		
80.45	10	93	0.117	74.9	9	150	0.231	85.36	9	54	0.169

Table 2: Summary of the results obtained for the two real data sets Iris and Iris23

Iris				**Iris23**			
th(%)	**c**	**t**	**XB**	**th(%)**	**c**	**t**	**XB**
17.45	**2**	14	**0.054**	11.65	**2**	24	**0.158**
47.45	3	33	0.137	49.65	3	41	0.195
62.45	4	26	0.632	65.15	4	42	0.210
72.45	5	44	0.412	65.65	5	84	0.457
73.45	6	32	0.311	70.65	6	139	0.276
74.45	7	68	0.374	71.65	7	67	0.361
75.45	8	59	0.254	72.65	8	63	0.335
76.45	9	52	0.358	73.65	9	208	0.769
79.45	10	48	0.323	77.65	10	102	0.455

Table 3: Actual centers, learned prototypes, and misclassification errors produced by 1-NP classifier

Data set	**Actual centres**	**Learned prototypes**	**Percentage of misclassification errors after 1-NP**
X1	$\begin{pmatrix} 1.574 & 3.636 \\ 1.647 & 1.729 \end{pmatrix}$	$\begin{pmatrix} 1.602 & 3.692 \\ 1.654 & 1.726 \end{pmatrix}$	0%
X2	$\begin{pmatrix} 07.350 & 25.950 \\ 18.250 & 19.000 \end{pmatrix}$	$\begin{pmatrix} 07.522 & 27.073 \\ 18.519 & 18.699 \end{pmatrix}$	10%
X3	$\begin{pmatrix} 3.847 & 1.061 & 4.076 \\ 0.574 & 3.973 & 4.235 \end{pmatrix}$	$\begin{pmatrix} 3.846 & 1.078 & 4.090 \\ 0.569 & 3.980 & 4.254 \end{pmatrix}$	0%
Iris23	$\begin{pmatrix} 5.936 & 6.588 \\ 2.770 & 2.974 \\ 4.260 & 5.552 \\ 1.326 & 2.026 \end{pmatrix}$	$\begin{pmatrix} 5.920 & 6.597 \\ 2.779 & 2.996 \\ 4.409 & 5.418 \\ 1.415 & 1.958 \end{pmatrix}$	15%

Finally, to show the ability of the proposed method to characterise the detected classes in terms of prototypes, Table 3 displays, for the four examples whose number of classes has been correctly determined, both the actual centres (calculated from original labelled data), and those produced by the algorithm. We can see that the learned prototypes are very close to the actual centres and, therefore, can be utilised as representatives of the corresponding classes. To test the usefulness of these representatives, we used them as a basis for the nearest prototype (1-NP) classifier [16] in order to classify all objects of each data set. The results of this application are reported on the last column of Table 3 in terms of percentage of misclassification errors.

5.0 CONCLUSION

The method proposed in this paper is an attempt to dim one of the most difficult problems facing clustering algorithms which is the problem of automatic determination of the number of clusters. This method is mainly based on an unsupervised fuzzy learning procedure that rapidly explores the data in order to discover the clusters they form. In addition to the number of clusters, this procedure provides a matrix of prototypes that can be optimised using the FCM algorithm.

The number of detectable clusters can be influenced by an algorithmic parameter, th, which represents the minimum of similarity each element should have with its nearest prototype. In practice, if no prior information can justify the use of a particular value of th, the learning process is repeated for different values comprising between two limits automatically determined from the data at hand. This means that more than one candidate solution can be

produced for each learning database. To select the most appropriate one among these solutions, a validity process is utilised.

The usefulness of this method has been illustrated on both artificial and real test data; and the obtained results encourage future investigations in order to improve the method by incorporating, for example, new similarity measures, new learning protocols, and/or new optimisation techniques. It would be also interesting to find a way to adapt the similarity threshold to each class, instead of using the same threshold for all classes.

REFERENCES

[1] J.C. Bezdek, "A Review of Probabilistic, Fuzzy, and Neural Models for Pattern Recognition". *Journal of Intelligent and Fuzzy Systems*, Vol. 1 No. 1, 1993, pp. 1-25.

[2] T. Pavlidis, "36 years on the Pattern Recognition Front. Lecture given at ICPR'2000 in Barcelona, Spain on the occasion of receiving the K.S. Fu prize". *Pattern Recognition Letters*, Vol. 24 No. 1-3, 2003, pp. 1-7.

[3] A. K. Jain, M. N. Murty and P. J. Flynn, "Data Clustering: A Review". *ACM Computing Surveys*, Vol. 31 No. 3, 1999, pp. 264-323.

[4] R. O. Duda, P. E. Hart and D. J. Stork, *Pattern Classification*. 2nd ed., New York, Wiley-Interscience, 2000.

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

[6] A. Bouroumi, M. Limouri and A. Essaïd, "Unsupervised Fuzzy Learning and Cluster Seeking". *Intelligent Data Analysis*, Vol. 4 No. 3, 2000, pp. 1-13.

[7] L. A. Zadeh, "Fuzzy Sets". *Inform. Contr.* Vol. 8, 1965, pp. 338-353.

[8] M. J. Sabin, "Convergence and Consistency of Fuzzy c-means/ISODATA Algorithms". *IEEE Trans. Pattern Anal. Machine Intell.* Vol. 9 No. 5, 1987, pp. 661-668.

[9] A. M. Bensaid, L. O. Hall, J. C. Bezdek, L. P. Clarke and M. L. Silbeger, "Validity-Guided (Re)Clustering with Applications to Image Segmentation". *IEEE Trans. Fuzzy Systems*. Vol. 4 No. 2, 1996, pp. 112-123.

[10] N. B. Karayiannis, J. C. Bezdek, N. R. Pal, R. J. Hathaway and P. I. Pai, "Repairs to GLVQ: A New Family of Competitive Learning Schemes". *IEEE Trans. On Neural Networks*. Vol. 7 No. 5, 1996, pp. 1062-1071.

[11] A. El Imrani, A. Bouroumi, M. Limouri and A. Essaïd, "A Coevolutionary Genetic Algorithm Using Fuzzy Clustering". *Intelligent Data Analysis*. Vol. 4 No. 3, 4, 2000, pp. 183-193.

[12] X. L. Xie and G. A. Beni, "Validity Measure for Fuzzy Clustering". *IEEE Trans. Pattern Anal. Machine Intell.* Vol. 3 No. 8, 1991, pp. 841-846.

[13] N. R. Pal and J. C. Bezdek, "On Cluster Validity for the Fuzzy c-Means Model". *IEEE Trans. Fuzzy Systems*. Vol. 3 No. 3, 1995, pp. 370-379.

[14] D. W. Kim, K. H. Lee and D. Lee, "Fuzzy Cluster Validation Index Based on Inter-Cluster Proximity". *Pattern Recognition Letters*. Vol. 24, 2003, pp. 2561-2574.

[15] J. C. Bezdek, M. Keller, R. Krishnapuram, L. I. Kuncheva, and N. R. Pal, "Will the Real Iris Data Please Stand up?" *IEEE trans. Fuzzy Systems*. Vol. 7 No. 3, 1999, pp. 368-369.

[16] L. I. Kuncheva and J. C. Bezdek, "Nearest Prototype Classification: Clustering, Genetic Algorithms, or Random Search?" *IEEE Trans. Syst., Man, Cyber. Part C*. Vol. 28 No. 1, 1998, pp. 160-164.

BIOGRAPHY

Mohamed Benrabh is an Assistant Professor in the Faculty of Sciences, Hassan II-Mohammedia University, Casablanca, Morocco. The main fields of his research interests concern intelligent systems and learning algorithms.

Abdelaziz Bouroumi is a Professor in the Faculty of Sciences, Hassan II-Mohammedia University, Casablanca, Morocco. His current research interests include fuzzy logic, neural networks, and evolutionary algorithms.

Abdellatif Hamdoun is a Professor in the Faculty of Sciences, Hassan II-Mohammedia University, Casablanca, Morocco, where he founded, in the early nineties of the last century, the Information Processing Laboratory. He is interested in the design of automatic systems for data acquisition and information processing.