ADAPTATION OF EXCLUSION/INCLUSION HYPERBOXES

FOR CLASSIFICATION OF COMPLEX DATA

Andrzej Bargiela [1], Masahiro Tanaka [2] , Giovanna Castellano [3], Annamaria Fanelli [3]

[1] *The Nottingham Trent University, Nottingham NG1 4BU, UK*

[2] *Konan University, 8-9-1 Okamoto, Higashinada-ku, Kobe, Japan*

[3] *University of Bari, via Orabona 4, 70126 Bari, Italy*

andre@doc.ntu.ac.uk, *m_tanaka@konan-u.ac.jp*, *castellano@di.uniba.it*, *fanelli@di.uniba.it*

ABSTRACT

Exclusion/inclusion hyperbox classification has demonstrated significant advantages in terms of its ability to cover topologically complex data structures with a relatively few hyperboxes thus resulting in the superior interpretability of classification results. However, the size of exclusion hyperboxes may occasionally become prohibitive if the data classes are grouped in a particularly unfavorable way in the pattern space. In this study we consider adaptation of the maximum size of hyperboxes in response to the ratio of the exclusion to inclusion hyperboxes. Two alternative adaptation strategies are being considered: (i) the adaptation of the size of all hyperboxes and (ii) the adaptation of the size of hyperboxes that fall within the previously identified exclusion area. The tradeoff between the number and the complexity of the classification rules implied by the two strategies is assessed on a set of sample classification problems.

Key Words: Pattern classification, exclusion/inclusion hyperboxes, min-max neural networks, information granulation, Information and knowledge management

1. INTRODUCTION

The use of fuzzy sets for the representation of real-life data has been proposed by Zadeh (1965) who pointed out that typically real-life data is not *crisp* but is characterized by a *degree of membership*. In this case the use of traditional set theory forces unrealistic binary classification decisions where the graded response is more appropriate. An early application of fuzzy sets to the pattern classification problem (Bellmann et al, 1966) proves the point that fuzzy sets represent an excellent tool simplifying the representation of complex boundaries between the pattern classes while retaining the full expressive power for the representation of the *core area* for each class. By having classes represented by fuzzy set membership functions it is possible to describe the degree to which a pattern belongs to one class or another.

Bearing in mind that the purpose of classification is the enhancement of interpretability of data or, in other words, derivation of a good abstraction of such data the use of hyperbox fuzzy sets as a description of pattern classes provides clear advantages. Each hyperbox can be interpreted as a fuzzy rule. However, the use of a single hyperbox fuzzy set for each pattern class is too limiting in that the topology of the original data is frequently quite complex (and incompatible with the convex topology of the hyperbox). This limitation can be overcome by using a collection (union) of hyperboxes to cover each pattern class set (Simpson, 1992, 1993)

(Gabrys et al, 2000). Clearly, the smaller the hyperboxes the more accurate cover of the class set can be obtained. Unfortunately, this comes at the expense of increasing the number of hyperboxes, thus eroding the original objective of interpretability of the classification result. We have therefore a task of balancing the requirements of accuracy of coverage of the original data (which translates on the minimization of misclassifications) with the interpretability of class sets composed of many hyperboxes.

The solution originally proposed by Simpson (1992) was the optimization of a single parameter defining the maximum hyperbox size as a function of misclassification rate. However, the use of a single maximum hyperbox size is somewhat restrictive. For class sets that are well separated from each other the use of large hyperboxes is quite adequate while for the closely spaced class sets, with a complex partition boundary, there is a need for small hyperboxes, so as to avoid high misclassification rates. A more general solution proposed in (Gabrys et al, 2000), involved the adaptation of the size of individual hyperboxes so that it is possible to generate larger hyperboxes in some areas of the pattern space while in the other areas the hyperboxes are constrained to be small to maintain low misclassification rates. The adaptation procedure requires however several presentations of data to arrive at the optimum sizes of hyperbox sizes for the individual classes.

The above two approaches both generate class sets as a union of hyperbox sets. A different approach that expresses class sets as a difference of two fuzzy sets has been proposed in (Bargiela et al, 2003). In this approach, the first set that is generated is a union of hyperboxes produced in the standard way (inclusion set) and the second set is a union of intersections of all hyperboxes that belong to different classes (exclusion set). By subtracting the exclusion hyperboxes from the inclusion ones it is possible to express complex topologies of the class set using fewer hyperboxes. Also, the three steps of the Min-Max clustering (Simpson, 1992; Gabrys et al, 2000) namely *expansion*, *overlap test* and *contraction* can be reduced to two: *expansion* and *overlap* tests.

This paper builds on the result reported in (Bargiela et al, 2003) and explores the adaptation of the maximum hyperbox size as a function of the ratio of exclusion and inclusion hyperboxes. Section 2 gives an overview of the fuzzy Min-Max classification algorithm. In Section 3 we discuss problems inherent to the Min-Max algorithm and describe the exclusion-inclusion fuzzy hyperbox classification algorithm. Section 4 describes the proposed adaptation algorithm and Section 5 provides numerical examples.

2. FUZZY MIN-MAX CLASSIFICATION

The fuzzy Min-Max classification neural networks are built using hyperbox fuzzy sets. A hyperbox defines a region in $\mathbf{R}^n$, or more specifically in $[0\ 1]^n$ (since the data is normalized to [0 1]) and all patterns contained within the hyperbox have full class membership. A hyperbox $\mathbf{B}$ is fully defined by its minimum $\mathbf{V}$ and maximum $\mathbf{W}$ vertices. So that, $\mathbf{B}=[\mathbf{V}, \mathbf{W}] \subset [0\ 1]^n$ with $\mathbf{V}, \mathbf{W} \in [0\ 1]^n$.

Fuzzy hyperbox $\boldsymbol{B}$ is described by a membership function (in addition to its minimum and maximum vertices), which maps the universe of discourse ($\mathbf{X}$) into a unit interval

$$\boldsymbol{B} : \mathbf{X} \rightarrow [0, 1] \tag{1}$$

Formally, $\boldsymbol{B}$(x) denotes a degree of membership that describes an extent to which x belongs to $\boldsymbol{B}$. If $\boldsymbol{B}$(x) =1 then we say that x fully belongs to $\boldsymbol{B}$. If $\boldsymbol{B}$(x) is equal to zero, x is fully excluded from $\boldsymbol{B}$. The values of the membership function that are in-between 0 and 1 represent a partial membership of x to $\boldsymbol{B}$. The higher the membership grade, the stronger is the association of the given element to $\boldsymbol{B}$. In this paper we will use an alternative notation for the hyperbox membership function $b(\boldsymbol{X}, \mathbf{V}, \mathbf{W})$ which gives an explicit indication of the min- and max- points of the hyperbox. The hyperbox fuzzy set will then be denoted as B={$\boldsymbol{X}$, **V**, **W**, b($\boldsymbol{X}$, **V**, **W**)}. Note that $\boldsymbol{X}$ is an input pattern that in general represents a class-labelled hyperbox in $[0\ 1]^n$. To put it formally

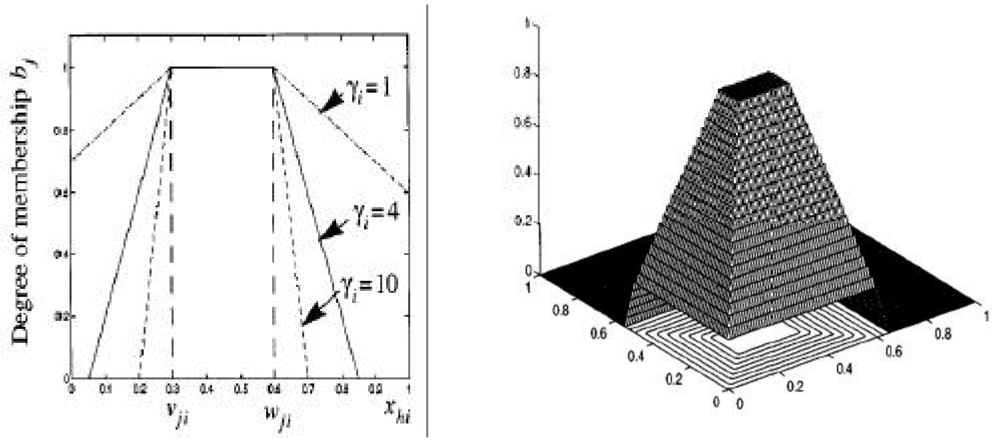
$$\boldsymbol{X}=\{[\boldsymbol{X}^l\ \boldsymbol{X}^u],\ \mathrm{d}\} \tag{2}$$

where $\boldsymbol{X}^l$ and $\boldsymbol{X}^u$ represent min and max points of the input hyperbox $\boldsymbol{X}$ and $\mathrm{d}\in\{1,\ldots,p\}$ is the index of the classes that are present in the data set.

While it is possible to define various hyperbox membership functions that satisfy the boundary conditions with regard to full inclusion and full exclusion, it is quite intuitive to adopt a function that ensures monotonic (linear) change in-between these extremes. Following the suggestion in (Sipmson, 1992) we adopt here

$$b_j(X_h) = \min_{i=1,\ldots,n}(\min([1-f(x_{hi}^u - w_{ji}, g_i)],\ [1-f(v_{ji}-x_{hi}^l, g_i)])) \tag{3}$$

where $f(r,g)=\begin{cases}1 & if \quad rg>1\\ rg & if \quad 0\le rg\le 1\\ 0 & if \quad rg<0\end{cases}$ is a two parameter function in which r represents the distance of the test pattern X_h from the hyperbox [$\boldsymbol{V}$ $\boldsymbol{W}$] and $g=[g_1, g_2, \ldots, g_n]$ represents the gradient of change of the fuzzy membership function. This is illustrated in Figure 1.



(a) (b)

Figure 1. One-dimensional (a) and two-dimensional (b) fuzzy membership function evaluated for a point input pattern $\boldsymbol{X}_h$.

The fuzzy Min-Max algorithm is initiated with a single point hyperbox [$\boldsymbol{V}_j$ $\boldsymbol{W}_j$]=[$\boldsymbol{0}$ $\boldsymbol{0}$]. However, this hyperbox does not persist in the final solution. As the first input pattern $\boldsymbol{X}_h$={[$\boldsymbol{X}_h^l$ $\boldsymbol{X}_h^u$], d} is presented the initial hyperbox becomes [$\boldsymbol{V}_j$ $\boldsymbol{W}_j$]= [$\boldsymbol{X}_h^l$ $\boldsymbol{X}_h^u$]. Presentation of subsequent input patterns has an effect of creating new hyperboxes or modifying the size of the existing ones. A special case occurs when a new pattern falls inside an existing hyperbox in which case no modification to the hyperbox is needed.

Hyperbox expansion: When the input pattern $\boldsymbol{X}_h$ is presented the fuzzy membership function for each hyperbox is evaluated. This creates a preference order for the inclusion of $\boldsymbol{X}_h$ in the existing hyperboxes. However the inclusion of the pattern is subject to two conditions: (a) the new pattern can only be included in the hyperbox if the class label of the pattern and the hyperbox are the same and (b) the size of the expanded hyperbox that includes the new pattern must not be greater in any dimension than the maximum permitted size. To put it formally the expansion procedure involves the following

$$if\ class(B_j) = \begin{cases} d_h \ \Rightarrow & test\ if\ B_j\ satisfies\ the\ \max imum\ size\ constra\mathrm{int} \\ else\ \Rightarrow & take\ another\ B_j \end{cases} \tag{4}$$

with the size constraint in (4) defined as

$$\underset{i=1,\ldots,n}{\forall} (\max(w_{ji}, x_{hi}^u) - \min(v_{ji}, x_{hi}^l)) \le \Theta \tag{5}$$

If expansion can be accomplished then the hyperbox min and max points are updated as

$$v_{ji} = \min(v_{ji}, x_{hi}^l), \quad foreach\ i = 1,\ldots,n$$
$$w_{ji} = \max(w_{ji}, x_{hi}^u), \quad foreachi = 1,\ldots,n$$

The parameter Θ can either be a scalar, as suggested in [7], or a vector defining different maximum hyperbox sizes in different dimensions [4]. It can be shown that the latter can result in fewer hyperboxes defining each pattern class but requires some a-priori knowledge about the topology of individual class sets or multiple presentations of data to facilitate adaptation.

Overlap test: The expansion of the hyperboxes can produce hyperbox overlap. The overlap of hyperboxes that have the same class labels does not present any problem but the overlap of hyperboxes with different class labels must be prevented since it would create ambiguous classification. The test adopted in [7] and [4] adopts the principle of minimal adjustment, where only the smallest overlap for one dimension is adjusted to resolve the overlap. This involves consideration of four cases for each dimension

Case 1: $v_{ji} < v_{ki} < w_{ji} < w_{ki}$

Case 2: $v_{ki} < v_{ji} < w_{ki} < w_{ji}$

Case 3: $v_{ji} < v_{ki} < w_{ki} < w_{ji}$

Case 4: $v_{ki} < v_{ji} < w_{ji} < w_{ki}$

The minimum value of overlap is remembered together with the index i of the dimension, which is stored as variable Δ. The procedure continues until no overlap is found for one of the dimensions (in which case there is no need for subsequent hyperbox contraction) or all dimensions have been tested.

Hyperbox contraction: The minimum overlap identified in the previous step provides basis for the implementation of the contraction procedure. Depending on which case has been identified the contraction is implemented as follows:

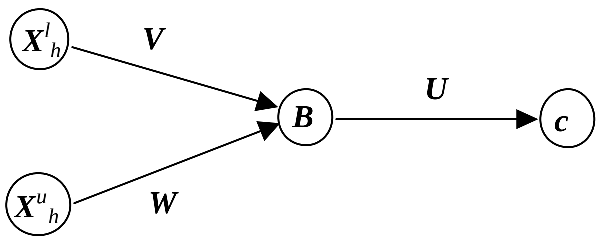
Case 1: $v_{k\Delta}^{new} = w_{j\Delta}^{new} = \dfrac{v_{k\Delta}^{old} + w_{j\Delta}^{old}}{2}$ $\quad oralternatively \ (w_{j\Delta}^{new} = v_{k\Delta}^{old})$

Case 2: $v_{j\Delta}^{new} = w_{k\Delta}^{new} = \dfrac{v_{j\Delta}^{old} + w_{k\Delta}^{old}}{2}$ $\quad oralternatively \ (v_{j\Delta}^{new} = w_{k\Delta}^{old})$

Case 3: $if \ \ w_{k\Delta} - v_{j\Delta} \le w_{j\Delta} - v_{k\Delta} \ then \ v_{j\Delta}^{new} = w_{k\Delta}^{old} \ otherwise \ w_{j\Delta}^{new} = v_{k\Delta}^{old}$

Case 4: $if \ \ w_{k\Delta} - v_{j\Delta} \le w_{j\Delta} - v_{k\Delta} \ then \ w_{k\Delta}^{new} = v_{j\Delta}^{old} \ otherwise \ v_{k\Delta}^{new} = w_{j\Delta}^{old}$

The above three steps of the fuzzy Min-Max classification can be expressed as training of a three-layer neural network. The network, represented in Figure 2, has a simple feed-forward structure and grows adaptively according to the demands of the classification problem. The input layer has $2*n$ processing elements, the first n elements deal with the min point of the input hyperbox and the second n elements deal with the max point of the input hyperbox X_h = $[X_h^l \ X_h^u]$. Each second-layer node represents a hyperbox fuzzy set where the connections of the first and second layers are the min-max points of the hyperbox including the given pattern and the transfer function is the hyperbox membership function. The connections are adjusted using the expansion, overlap test, contraction sequence described above. Note that the min points matrix **V** is modified only by the vector of lower bounds X_h^l of the input pattern and the max points matrix **W** is adjusted in response to the vector of upper bounds X_h^u.



$\boldsymbol{X}^l{}_h = [x^l{}_{h1}, \ldots, x^l{}_{hn}]$, $\boldsymbol{X}^u{}_h = [x^u{}_{h1}, \ldots, x^u{}_{hn}]$

$\boldsymbol{V}=[V_1,\ldots,V_m]$, $\boldsymbol{W}=[W_1,\ldots,W_m]$

$\boldsymbol{B}=[B_1,\ldots,B_m]$

$\boldsymbol{U}=[u_{11},\ldots,u_{1p};\ldots;\ u_{m1},\ldots,u_{mp}]$

$\boldsymbol{c}=[\ c_1,\ldots,c_p]$

Figure 2. The three-layer neural network implementation of the GFMM algorithm.

The connections between the second- and third-layer nodes are binary values. They are stored in matrix **U**. The elements of **U** are defined as follows:

$$u_{jk} = \begin{cases} 1 & if \ B_j \ is \ a \ hyperbox \ for class \ \ c_k \\ 0 & \qquad\qquad otherwise \end{cases} \tag{6}$$

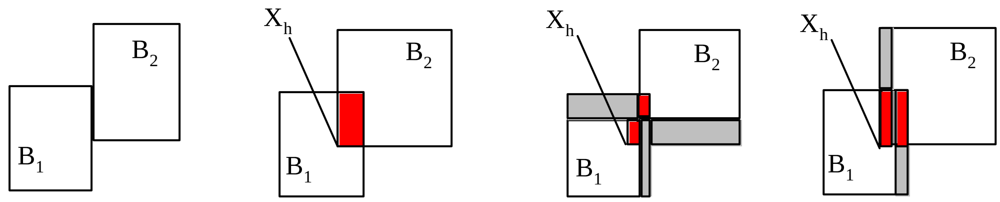
where B_j is the jth second-layer node and c_k is the kth third-layer node. Each third-layer node represents a class. The output of the third-layer node represents the degree to which the input pattern X_h fits within the class k. The transfer function for each of the third-layer nodes is defined as

$$c_k = \max_{j=1}^{m} B_j u_{jk} \tag{7}$$

for each of the p third-layer nodes. The outputs of the class layer nodes can be fuzzy when calculated using expression (7), or crisp when a value of one is assigned to the node with the largest c_k and zero to the other nodes.

3. EXCLUSION/INCLUSION CLASSIFICATION ALGORITHM

Training of the Min-Max neural network involves adaptive construction of hyperboxes guided by the class labels. The input patterns are presented in a sequential manner and are checked for a possible inclusion in the existing hyperboxes. If the pattern is fully included in one of the hyperboxes no adjustment of the min- and max-point of the hyperbox is necessary, otherwise a hyperbox *expansion* is initiated. However, after expansion is accomplished it is necessary to perform an *overlap test* since it is possible that the expansion resulted in some areas of the pattern space belonging simultaneously to two distinct classes, thus contradicting the classification itself. If the overlap test is negative, the expanded hyperbox does not require any further adjustment and the next input pattern is being considered. If, on the other hand, the overlap test is positive the hyperbox *contraction* procedure is initiated. This involves subdivision of the hyperboxes along one or several overlapping coordinates and the consequent adjustment of the min- and max-points of the overlapping hyperboxes. However, the contraction procedure has an inherent weakness in that it inadvertently eliminates from the two hyperboxes some part of the pattern space that was unambiguous while in the same time retaining some of the contentious part of the pattern space in each of the hyperboxes. This is illustrated in Figure 3.



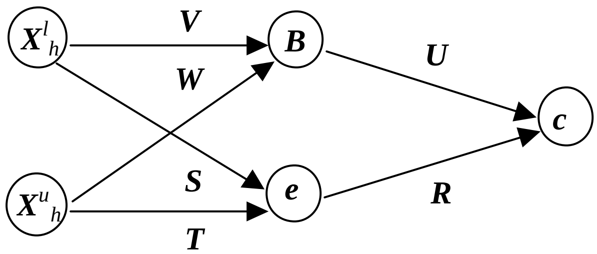
(a) (b) (c) (d)

Figure 3. Training of the fuzzy Min-Max neural network.
(a) Hyperboxes belonging to two different classes $class(B_1) \neq class(B_2)$;
(b) Inclusion of pattern $\{X_h, class(B_2)\}$ in B_2 implying overlap with B_1;
(c) Contraction of B_1 and B_2 with adjustment along two coordinates;
(d) Contraction of B_1 and B_2 with adjustment along one coordinate.

It is easy to see that the contraction step of the fuzzy Min-Max network training resolves only part of the problem created by the expansion of the hyperbox B_2. Although the hyperboxes B_1 and B_2 no longer overlap after the contraction has been completed (Figure 3(c) and 3(d)), some part of the original hyperbox B_1 remains included in B_2 and similarly some part of the hyperbox B_2 remains included in the contracted B_1. The degree of this residual inclusion depends on the contraction method that is chosen but it is never completely eliminated.

Another problem inherent to the contraction procedure is that it unnecessarily eliminates parts of the original hyperboxes. These eliminated portions are marked in Figure 3 with diagonal pattern lines. The elimination of these parts of hyperboxes implies that the contribution to the training of the Min-Max neural network of the data contained in these areas is nullified. If the neural network training involves only one pass through the data, then this is an irreversible loss of information that demonstrates itself in a degraded classification performance. The problem can be somewhat alleviated by allowing multiple presentations of data in the training process or reducing the maximum size of hyperboxes. In either case the result is that additional hyperboxes are created to cover the eliminated portions of the original hyperboxes. Also, it is worth noting that the training pattern $\{X_h, class(B_2)\}$ continues to be misclassified in spite of the contraction of the hyperboxes. This means that a 100% correct classification rate is not possible despite multiple-pass neural network training.

The exclusion/inclusion approach avoids the above problem by subtracting the overlapping area marked in red (Figure 3(b)) from each of the hyperbox B_1 and B_2. In this way the original hyperboxes do not lose any of the undisputed area of the pattern space and, in the same time, the patterns contained in the exclusion hyperbox are eliminated from the relevant classes in the set $\{c_1,\ldots,c_p\}$ and are assigned to class c_{p+1} (contentious area of the pattern space class). It is worth noticing that the subtraction of the exclusion hyperbox from hyperboxes B_1 and B_2 produces a convex shape for each of the classes in a very efficient way (in terms of the number of hyperboxes involved). The corresponding neural network that implements the exclusion/inclusion algorithm is presented in Figure 4.



$$\boldsymbol{X}^l_h = [X^l_{h1}, \ldots, X^l_{hn}],\ \boldsymbol{X}^u_h = [X^u_{h1}, \ldots, X^u_{hn}]$$
$$\boldsymbol{V}=[v_1,\ldots,v_m],\ \boldsymbol{W}=[w_1,\ldots,w_m],\ \boldsymbol{S}=[s_1,\ldots,s_m],\ \boldsymbol{T}=[t_1,\ldots,t_m]$$
$$\boldsymbol{B}=[B_1,\ldots,B_m],\ \boldsymbol{e}=[e_1,\ldots,e_q]$$
$$\boldsymbol{U}=[u_{10},\ldots, u_{1p}, u_{1(p+1)};\ldots; u_{m0},\ldots,u_{mp}, u_{m(p+1)}]$$
$$\boldsymbol{R}=[r_{10},\ldots, r_{1p}, r_{1(p+1)};\ldots; r_{q0},\ldots,r_{qp}, r_{q(p+1)}]$$
$$\boldsymbol{c}=[c_1,\ldots,c_p,c_{p+1}]$$

Figure 4. Exclusion/Inclusion Fuzzy Classification Network.

The additional second-layer nodes ***e*** are formed adaptively in a similar fashion as for nodes ***B***. The min-point and the max-point of the exclusion hyperbox are identified when the overlap test is positive for two hyperboxes representing different classes. These values are stored as new entries in matrix ***S*** and matrix ***T*** respectively. If the new exclusion hyperbox contains any of the previously identified exclusion hyperboxes, the included hyperboxes are eliminated from the set ***e***. The connections between the nodes ***e*** and nodes ***c*** are binary values stored in matrix ***R***. The elements of ***R*** are defined as follows:

$$r_{lk} = \begin{cases} 1 & if\ e_l\ overlapped\ hyperbox\ of\ class\ \ c_k\ and\ 1 < k < p \\ 1 & if \quad k = p+1 \\ 0 & otherwise \end{cases} \tag{8}$$

Note that the third layer has $p+1$ nodes $[c_1,\ldots, c_p, c_{p+1}]$ with the node c_{p+1} representing the new exclusion hyperbox class. The output of the third-layer is now moderated by the output from the exclusion hyperbox nodes ***e*** and the values of matrix ***R.*** The transfer function for the third-layer nodes is defined as:

$$c_k = \max_{k=1}^{p+1}(\max_{j=1}^{m} b_j u_{jk} - \max_{i=1}^{q} e_i r_{ik}\ ) \tag{9}$$

The second component in (9) cancels out the contribution from the overlapping hyperboxes that belonged to different classes.

4. ADAPTATION OF HYPEERBOX SIZE

The exclusion/inclusion classification algorithm described above can be made fully data-driven i.e. the formation of inclusion and exclusion hyperboxes can proceed without reference to any external parameter such as a maximum hyperbox size. While this is in general a very welcome feature, it also means that for some "difficult" data sets the exclusion hyperboxes can become large relative to the inclusion hyperboxes. If this is the case the proportion of patterns that cannot be classified as belonging to a specific class may become unacceptably high. We therefore provide a feedback mechanism that rectifies this problem.

The proposed adaptive exclusion/inclusion classification algorithm can be formalized as follows:

- *Perform the exclusion/inclusion neural network training as described in Section 3 (without any constraint on the maximum size of hyperboxes;* $\Theta \le 1$*);*
- *Evaluate the volume of all exclusion and inclusion hyperboxes;*
- *If the ratio of volumes of exclusion to inclusion hyperboxes exceeds a pre-specified limit evaluate the parameter* Θ_x *as a product of the size of the maximum exclusion hyperboxes and a convergence parameter* a, $0 < a < 1$*;*
- *Repeat the exclusion/inclusion neural network training applying the constraint on the maximum size of hyperboxes* Θ_x *to those patterns that fall inside the exclusion hyperboxes identified in the previous iteration;*
- *Terminate when the ratio of exclusion to inclusion hyperboxes remains constant in two consecutive iterations.*

The choice of the parameter α enables a degree of control over the convergence rate of the algorithm. If α is small (e.g. 0.2), the convergence is rapid but the final value of the parameter Θ_x may be smaller than it is necessary to satisfy the requirement on the ratio of volumes of exclusion to inclusion hyperboxes. This means that the classification error may be larger than could be obtained with larger α. On the other hand, if the parameter α is large (e.g. 0.8) the convergence is correspondingly slower and the parameter Θ_x can be found more accurately. However, using larger α may lead to identical ratios of volumes of exclusion to inclusion hyperboxes in consecutive iterations thus terminating the algorithm prematurely. The consequence of that is that the number of patterns that cannot be classified (fall into exclusion hyperboxes) is larger than that obtainable with smaller α. We have found, through experimentation, that for many practical data classification problems the choice of α=0.5 results in good convergence, good classification accuracy and a small number of patterns that are not classified.

We illustrate these considerations using a synthetic data set with two overlapping data classes. Both training and test data sets are generated randomly and comprise of 100 data points in each class with patterns uniformly distributed around the points (0.4, 0.6) and (0.7, 0.3) respectively. A representative example of such data is given in Figure 5.

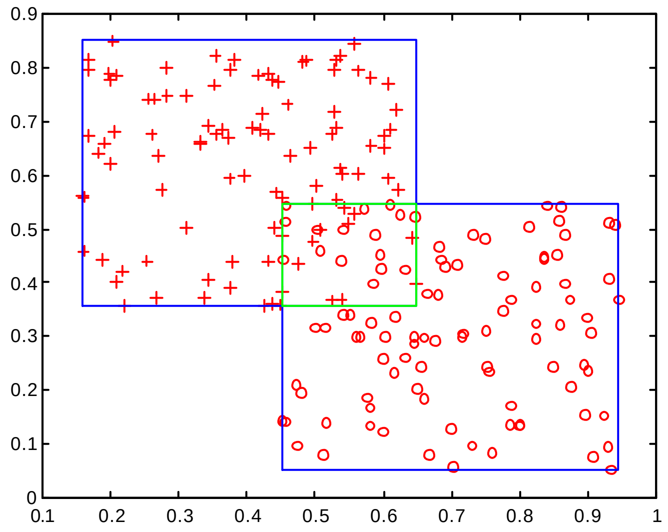


Figure 5. Example of a synthetic data set for the evaluation of the exclusion/inclusion classification algorithm.

The evaluation of the adaptive exclusion/inclusion classification algorithm is illustrated in Figures 6-9. In each figure runs 1-10 correspond to the parameter α=0.2, runs 11-20 correspond to α=0.4, runs 21-30 correspond to α=0.6 and runs 31-40 correspond to α=0.8. Four aspects are being assessed: the reduction of the ratio of the volume of exclusion to inclusion hyperboxes (Figure 6), the improvement of the classification accuracy (Figure 7), the reduction of the number of unclassified patterns (Figure 8) and the increase of the number of hyperboxes (Figure 9).

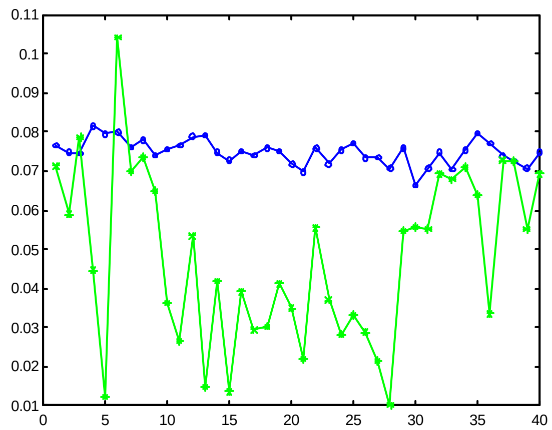


Figure 6. Reduction of the ratio of the volume of exclusion to inclusion hyperboxes (the ratio calculated originally for Θ =1 is plotted as blue “o” marks and the ratio obtained after adaptive shrinking of exclusion hyperboxes is plotted as green ‘*’ marks)

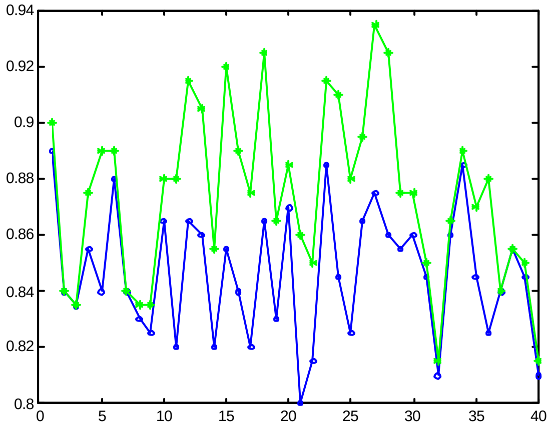


Figure 7. Improvement of the classification accuracy (the accuracy calculated originally for Θ =1 is plotted as blue “o” marks and the accuracy attained after adaptive shrinking of exclusion hyperboxes is plotted as green ‘*’ marks)

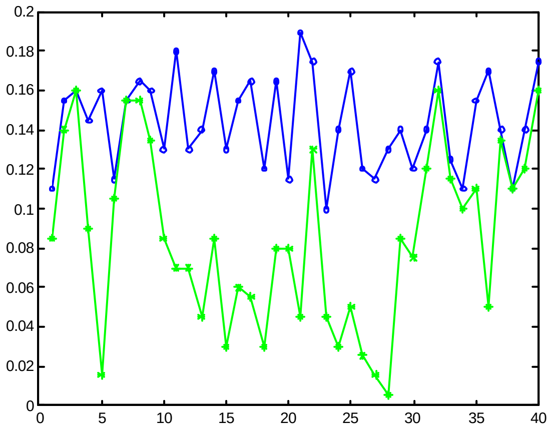


Figure 8. Reduction of the number of unclassified patterns (the original number of unclassified patterns for Θ =1 is plotted as blue “o” marks and the number of unclassified patterns after adaptive shrinking of exclusion hyperboxes is plotted as green ‘*’ marks)

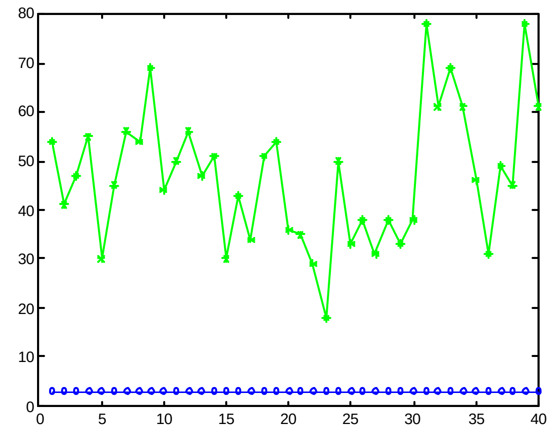


Figure 9. Increase of the number of hyperboxes (the original number of hyperboxes for Θ =1 is plotted as blue “o” marks and the number of hyperboxes after adaptive shrinking of exclusion hyperboxes is plotted as green ‘*’ marks)

It is clear from the above results that the adaptive exclusion/inclusion algorithm improves on the original exclusion/inclusion classification in all respects. With the new algorithm it is possible to reduce the ratio of volumes of exclusion to inclusion hyperboxes, improve the classification accuracy and reduce the number of patterns that cannot be classified. These improvements are obtained at the cost of increasing the overall number of hyperboxes. However, it is fortunate that the smallest increase of the number of hyperboxes occurs for the same range of the parameter α for which one can obtain the maximum benefits in terms of classification accuracy and the minimum number of unclassified patterns. Table 1 provides a summary of the results obtained for 10 runs with random data sets for each of the four values of the parameter α. The choice of α of around 0.6 gives the best balance between classification accuracy and the interpretability of results.

Table 1. Summary of the performance improvement due to the adaptive exclusion/inclusion classification

Shrinking coefficient	Ratio of new to old VolExcl/VolIncl	Ratio of new to old "correctly classified"	Ratio of new to old "excluded from classsification"	Ratio of new to old number of hyperboxes
0.2	0.7962	1.0141	0.7732	16.5000
0.4	0.4298	1.0557	0.4116	15.0667
0.6	0.4742	1.0513	0.3607	11.4333
0.8	0.8511	1.0131	0.8194	19.3000

In order to assess the value of the selective adaptation of hyperbox size described above we compare it with an alternative adaptation strategy that applies the same constraint on the maximum hyperbox size to all hyperboxes formed in consecutive iterations. The algorithm implementing this alternative strategy can be formalized as follows:

- *Perform the exclusion/inclusion neural network training as described in Section 3 (without any constraint on the maximum size of hyperboxes;* $\Theta \le 1$*);*
- *Evaluate the volume of all exclusion and inclusion hyperboxes;*
- *If the ratio of volumes of exclusion to inclusion hyperboxes exceeds a pre-specified limit reduce the value3 of the parameter* Θ *by multiplying it by a convergence parameter* a, $0 < a < 1$*;*
- *Repeat the exclusion/inclusion neural network training applying the constraint on the maximum size of hyperboxes* Θ*;*
- *Terminate when the ratio of exclusion to inclusion hyperboxes stops decreasing.*

The evaluation of this alternative classification algorithm follows the same procedure as described above. We perform 10 training/testing cycles on randomly generated data for each of the four values of the convergence parameter α. The results are summarized in Table 2 and indicate that although it is possible to improve the classification accuracy with the alternative adaptive exclusion/inclusion classification this is achieved at a cost of a significant increase of the number of hyperboxes. Consequently the interpretability of such classification is greatly impeded. For example for the value of a=0.6 the average number of hyperboxes created to cover the 200 patterns is 94 (31.333*3). This compares very unfavorably with only 34 hyperboxes generated by the algorithm that applies selectively the maximum hyperbox size constraint.

Table 2. Performance of the alternative adaptive exclusion/inclusion classification algorithm

Shrinking coefficient	Ratio of new to old VolExcl/VolIncl	Ratio of new to old "correctly classified"	Ratio of new to old "excluded from classsification"	Ratio of new to old number of hyperboxes
0.2	0.0296	1.0625	0	38.2000
0.4	0.0458	1.0743	0.0035	43.9000
0.6	0.0531	1.0914	0	31.3333
0.8	0.0695	1.0729	0.0148	26.4000

5. CONCLUSIONS

The paper presented a modification of our earlier exclusion/inclusion classification algorithm that balances the requirements of classification accuracy and interpretability of results. Two alternative algorithms for the adaptation of hyperbox size have been considered and the one based on selective application of the maximum hyperbox constraint has been found to produce the best results. The conclusions derived from the assessment of the algorithm on the randomly generated data set carry also to the standard classification problems such as defined by the *Idis data set*.

REFERENCES

Bargiela A., Pedrycz W., Granular clustering with partial supervision, *Proc. European Simulation Multiconference, ESM2001*, Prague, June 2001, 113-120.

Bargiela A., Pedrycz W., Tanaka M., Exclusion/Inclusion fuzzy classification network, *Knowledge Engineering Systems KES'2003*, Oxford, Sept. 2003.

Bellman, R.E., Kalaba, R., Zadeh, L., 1966, Abstraction and pattern classification, *J. Math. Anal. Appl.*, 13, 1-7

Dubois, D., Prade, H., Fuzzy relation equations and causal reasoning, *FSS*, 75, 1995, pp. 119-134

Gabrys B., Bargiela A., General fuzzy min-max neural network for clustering and classification, *IEEE Trans. Neural Networks*, vol.11, 2000, 769-783.

Pedrycz, W., 1992, Fuzzy neural networks with reference neurons as pattern classifiers, *IEEE Trans. Neural Networks*, 5, 770-775

Pedrycz, W., Gomide, F., *An Introduction to Fuzzy Sets*, Cambridge, MIT Press, Cambridge, MA, 1998.

Simpson PK., Fuzzy min-max neural networks - Part1: Classification, *IEEE Trans. Neural Networks*, vol.3, 5, 1992, 776-786.

Simpson PK., Fuzzy min-max neural networks – Part 2: Clustering, *IEEE Trans. Neural Networks*, vol.4, 1, 1993, 32-45.

Zadeh, L.A., Fuzzy sets, *Inform. And Control*, 8, 1965, 189-200.

Zadeh, L.A., Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems (FSS)*, 90, 1997, 111-117.