Fuzzy Multimodels

Witold Pedrycz
Department of Electrical and Computer Engineering, University of Manitoba
Winnipeg, Canada R3T 2N2
pedrycz@eeserv.ee.umanitoba.ca

Abstract Fuzzy multimodelling is concerned with the design and utilization of families rather than single models. The intent is to approximate data that are originated by phenomena whose nature is more relation - based than function - oriented. In general, fuzzy multimodels comprise a collection of local models $M_1, M_2, ..., M_c$ along with the relevant mechanisms of triggering and aggregating aimed at assuring a suitable interaction between these models. The idea of multimodelling is contrasted with some other approaches to fuzzy modelling available in the current literature. The algorithmic details are laid down and illustrated through several detailed simulation studies.

1. Introductory remarks

Most if not all models encountered in fuzzy modelling [10] are concerned with representing or approximating relationships between input and output variables in the form of some functional dependencies. The visible exception comes in the form of fuzzy relational models [5, 7, 8]. As the name stipulates, these models are oriented towards capturing the dependencies between the variables in the form of relations. There are, however, numerous situations in which function - oriented models or even relation - focused cannot be sufficient enough. The intent of this study is to come up with a new modelling framework termed fuzzy multimodelling. The objective of fuzzy multimodels is to deliver an environment assuring a successful interaction between several relational or functional constructs and allow for their efficient utilization.

In general, the multimodel can be thus regarded as a collection of models M_1, M_2, ..., M_c equipped with some mechanisms aimed at a relevant of triggering between the models or, if necessary, aggregating the results furnished by the individual models. It is worth contrasting the proposed approach with those existing in the literature and point out their main differences. The concept of "local" fuzzy models that is fundamental to the idea outlined in [9] has nothing to do with the aspect of multimodeling addressed in this study. Even though the architecture of the multimodel to be studied here relies on a sort of local models, the identification methodology is evidently very distinct. Moreover the switching mechanism considered in [9] is set up in such a way that only a single result becomes released by the entire model. In contrast, as this will be clarified in depth, we allow for several models to be simultaneously invoked and actively pursued. Similarly, the other concept of regression clustering studied in [3] looks into certain related aspects but does not tackle explicitly the facet of multimodelling on which the entire discussion in this paper is dwelled upon.

Two examples below give a useful insight into the very essence of the discussed multimodelling problem and highlight an importance of an operational continuum of modelling that is spread between functions and relations.
Example 1. Consider a series of data sets given in Fig.1 (a) - (d). It is evident that in Fig 1(a) the relationships between x and y constitute evidently a "pure" function - there seem to be very few noisy data points and they could be easily identified as genuine outliers. In Fig. 1(c) we are faced with more difficult and conceptually intriguing question. One can still attempt to build a functional form of dependencies between x and y by treating the region W as the one containing points corrupted by a high level of noise. But is this fully legitimate ? It could well be that in this region one should rather structure the entire model in terms of a *relation* rather than a *function* between x and y. Similarly, in Fig. 1(b) we are again faced with relations - in fact M_2 and M_3 treated together form a relation. Roughly speaking, by admitting the language of fuzzy relations one captures some structural noise of a nonprobabilistic nature. The data in Fig.1(d) are definitely represented as a relation (what is shown here is known as a so-called s-curve in chemical engineering portraying a relation between temperature and a Damköhler number, d). The data can be split into three separate segments in the sequel giving rise to three models M_1, M_2, and M_3. Again the data in Fig.1(c) exhibit a quite high variety as we can distinguish between a function (model M_1 and M_3) and one apparent fuzzy relation (M_2). Whether the form of the last model (M_2) is functional or relational, it depends on the level of the structural noise one is willing to accept. If the answer is affirmative then M_2 can still be regarded as function (not a relation).

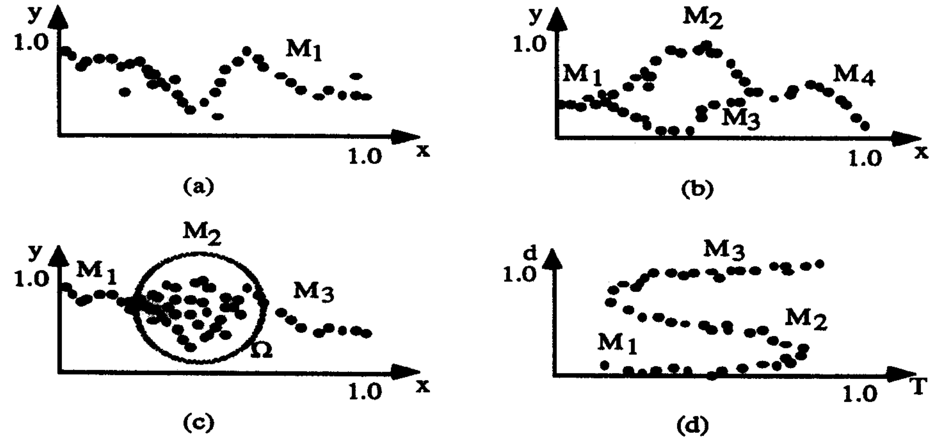

Fig. 1 Example data sets and their corresponding functional and relational models

The next example shows two situations in which the use of fuzzy models becomes even more indispensable.
Example 2. Obstacle avoidance. An autonomous robot is faced with an obstacle and has to avoid it. Not knowing the details of the navigation algorithm exercised by the robot, our intent is to build a respective model of obstacle avoidance by monitoring its behavior. To accomplish this we record a distance of the robot from the obstacle and the corresponding angle of a turn taken at this location, Fig. 2(a). The example log of these data is portrayed in Fig. 2(b). It is obvious that being rather close to the obstacle the robot takes a sharp turn right *or* left, see Fig.2 (b). Definitely, as the data are highly relational, the multimodel facet of modelling becomes highly required.

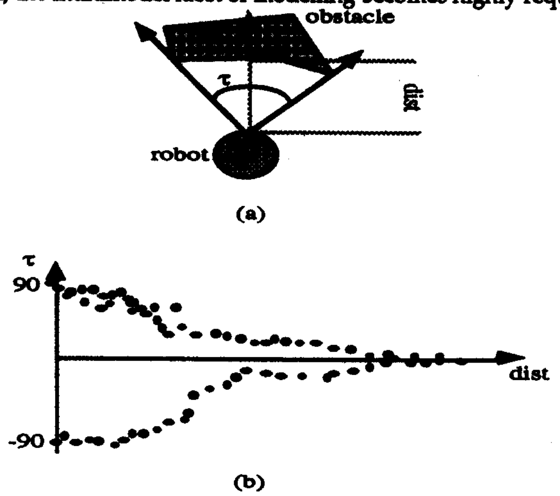

Fig. 2 Obstacle avoidance problem: (a) robot and obstacle configuration (b) distance - turning angle dependency

All the above problems constitute a genuine conceptual challenge to standard modelling techniques as being inherently relational. It is not the form of the model to be used that really matters but a way in which they are constructed and put up together. Imagine, for instance, the use of a single neural network, no matter how complex, in describing the above problem of obstacle avoidance. This will lead to a perfect averaging of the data (with subsequent highly undesired effects for the autonomous robot).

Having outlined the main ideas and motivation behind fuzzy multimodels, we move on with the general architecture of the multimodel and discuss pertinent algorithms.

2. The architecture of the mutlimodel

The main idea is to treat the multimodel as a collection of interacting models with the topology outlined in Fig. 3.

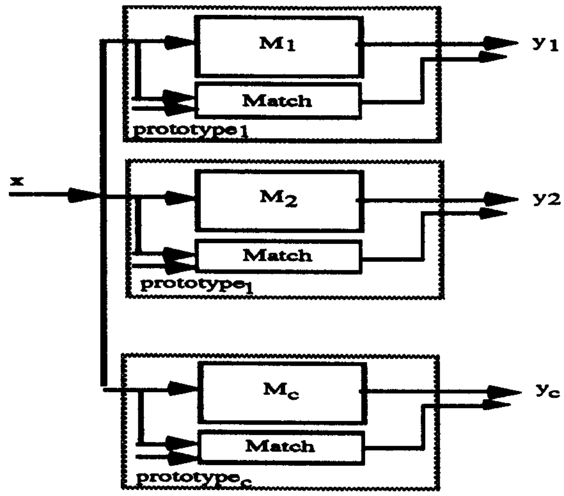

Fig. 4. General architecture of multimodelling

For completeness of discussion we assume that $x \in [0,1]^n$ and one-dimensional output $y \in [0,1]$. The input datum x is accepted by each model which in turn determine the corresponding output y_1, y_2, .., y_c. Note that we have not specified any particular form of the model; in fact one should stress that the discussed concept is model - independent. If necessary, at the later stage of the multimodel development each of the models M_i, i=1, 2, ..., c can be particularized as a fuzzy relational equation, linear or polynomial function, neural network, etc. With each model associated is a matching module (match) whose role is to express to which extent x should be handled by the i-th model. In other words the outcome of this matching reflects a degree of confidence to be associated with the result produced by the corresponding model. For instance, if there is only a single dominant level of matching g linked with the i_0 -th model, then this model does count and the result one should rely on is the one provided by it. On the other hand, if two or more models return similar levels of matching then the results of the corresponding models need to be considered - in this case the effect of multimodelling becomes evident as the output comes in the form of a collection of several admissible values, say y_{i_1}, y_{i_2}, y_{i_p}. The number of the admitted alternatives (the models taken into account) can be easily controlled by the predefined level of matching where the pertinent rule of model participation reads as

$$\text{- consider model } M_i \text{ if } g_i \geq g^*$$

where g^* is a certain threshold (called admission level) and g_i denotes the degree of confidence achieved for the i-th model, g_i, $g^* \in [0,1]$.

Noting that even though the selection of g^* is not overly critical still some caution should be exercised: too low g^*s can lead to a fairly numerous collection of the output results (as too many models get involved). On the other hand by moving the value of g^* too high the multimodel could easily become incomplete as none of its contributing models can be regarded relevant for some regions of the input variable(s). The selection of the threshold level constitutes an usual problem when compromising between specificity and completeness encountered in many other decision - making situations.

3. Development of fuzzy multimodels

In what follows we describe the construction of the fuzzy multimodel. Before getting into details, it is important to note that multimodels heavily exploit a concept of "locality" of their individual models. From the operational standpoint, it is essential to elaborate on how the partition of the space of the input variables can be carried out efficiently. The main idea is to reveal the structural relationship between the input and output variables via a method of specialized fuzzy clustering.

3.1. Fuzzy clustering and its directionality aspects

As the data using which the fuzzy multimodel is developed constitute collectively elements (vectors) in an (n+1) dimensional space, fuzzy clustering can be regarded as a suitable tool aimed at revealing a structure within the given data set. In contrast with the standard formulations of most of the currently available clustering problems [1, 2, 4], here we are interested in a direction of the dependencies in the data that is evidently from x to y (but not the other way around).

3.1.1. Performance index

As usual in any clustering method, we consider a data set consisting of ordered pairs (x_k, y_k), $k = 1, 2, \ldots, N$. where $x_k \in [0, 1]^n$ and $y_k \in [0, 1]$. The formation of the objective function (clustering criterion) used to describe an extent to which two pairs of data (x_k, y_k) and (x_l, y_l) could be regarded as the elements of the same cluster should be guided by the following commonsense observations,

(i) (x_k, y_k) and (x_l, y_l) treated as two candidates to be included in the same cluster should be similar coordinatewise, namely the corresponding coordinates of x_k and x_l as well as y_k and y_l should be *similar*,

(ii) the "directionality" component of the performance index should reflect the functional direction needed to be discovered within the data (meaning that " x implies y"); this fact should be reflected by the character of the elements assigned to the cluster. In such a sense, with x_k almost equal to x_l but quite different y_k and y_l, these two data should not be placed in the same cluster. On the other hand when x_k and x_l do not differ very significantly while having similar y_k and y_l, there is a high likelihood that these patterns could be allocated to the same cluster.

Let us first explicitly express how to quantify the notion of similarity. Since x_k and y_k are just fuzzy sets defined in the corresponding finite-dimensional spaces, one can adopt the equality index as originally defined in [6]. Recall briefly that the membership values "a" and "b" match to the degree $a \equiv b$ equal to

$$a \equiv b = \frac{1}{2}\left[\min(a \rightarrow b, b \rightarrow a) + \min(\bar{a} \rightarrow \bar{b}, \bar{b} \rightarrow \bar{a})\right] \tag{1}$$

where $a, b \in [0, 1]$, while $\rightarrow$ denotes a multivalued implication (pseudocomplement, cf. [7]), $a \rightarrow b = \{c \in [0,1] \mid a t c \leq b\}$, and the overbar symbol stands for the complement,

$\bar{a} = 1 - a$. Interestingly enough, the equality index equals 1 if and only if the two arguments (membership values) are the same, a=b. In particular, the implication can be specified in many different ways including several specific cases such as

Lukasiewicz implication

$$a \rightarrow b = \begin{cases} 1, & \text{if } a \leq b \\ 1-a+b, & \text{otherwise} \end{cases} = \min(1, 1-a+b) \tag{2}$$

Gödel implication

$$a \rightarrow b = \begin{cases} 1, & \text{if } a \leq b \\ b, & \text{otherwise} \end{cases} \tag{3}$$

Gaines

$$a \rightarrow b = \begin{cases} 1, & \text{if } a \leq b \\ \frac{b}{a}, & \text{otherwise} \end{cases} = \min(1, \frac{b}{a}) \tag{4}$$

$a, b \in [0,1]$.

The multidimensional version of (1) could be easily derived by averaging the results of the coordinatewise comparison obtained for the individual coordinates of x_k and x_l. This yields

$$x_k \equiv x_l = \frac{1}{n}\sum_{i=1}^{n} (x_{ki} \equiv x_{li}) \tag{5}$$

The necessary directionality aspect of the objective function is captured in the form

$$\left(y_k \equiv y_l\right) \rightarrow \left(x_k \equiv x_l\right) \tag{6}$$

As the clusters should be formed on a basis of both (i) and (ii), we aggregate (1) and (6) in a multiplicative way that gives rise to the formula,

$$Q = \frac{1}{2}\left[\left(x_k \equiv x_l\right) + \left(y_k \equiv y_l\right)\right]\left[\left(x_k \equiv x_l\right) \rightarrow \left(y_k \equiv y_l\right)\right] \tag{7}$$

that from now on will be used as the clustering objective function.

3.1.2. The clustering algorithm

The clustering procedure guided by the objective function (7) is applied successively to the individual elements of the data set. The process is carried out bottom-up in an aglomerative fashion: we start with N clusters each consisting of a single pair of the input-output patterns and merge them successively based on the values of the objective function produced via this combination. Starting from "N" single-element clusters $\{x_1, y_1)\}$ $\{(x_2, y_2)\}$, ..., $\{(x_N, y_N)\}$, the new two-element cluster

$$\{(x_{i0}, y_{i0}), (x_{j0}, y_{j0})\}$$

is formed in such a way it achieves a maximum of Q determined over all possible mergings of the available data points. Subsequently, the clusters to be expanded (merged) at the successive stages are guided by the maximal value of the performance index Q averaged over the corresponding cluster. This leads to the general merging rule:

- merge clusters X and X' for which the sum

$$\frac{1}{\mathrm{card}(\mathcal{X})\,\mathrm{card}(\mathcal{X}')} \sum_{(x_k, y_k) \in \mathcal{X}} \; \sum_{(x_l, y_l) \in \mathcal{X}'} Q(k, l) \tag{8}$$

attains a maximal value among all possible merging options.

The fundamental question that usually occurs when using any clustering technique is the one about a "plausible" number of the clusters to be distinguished in the data set. Since the proposed method is of an agglomerative nature, one should be able to control the process of merging by terminating it when the produced clusters cannot sufficiently represent the data. The deficient representation phenomenon may occur due to an excessive variety of the objects placed within the same cluster. This in turn calls for a formal definition of the representation capabilities of the clusters. More precisely, we will be interested in expressing how well the prototypes represent the elements of the generated clusters. Let us introduce the following notion. A prototype

$$p = (p_x, p_y) \in [0, 1]^n \times [0, 1]$$

of cluster X is taken as one of its elements

$$(x_{i0}, y_{i0})$$

such that it maximizes the objective function (7) where the computations concern all the elements belonging to this cluster

$$p_x = x_{i0} \text{ and } p_y = y_{i0} \text{ if } \max_{(x_l, y_l)} \sum_{k=1}^{\mathrm{card}(\mathcal{X})} Q(k, l) = \sum_{k=1}^{\mathrm{card}(\mathcal{X})} Q(k, i_0) \tag{9}$$

(note that the indices in Q are used to emphasize the data being discussed).

The resulting value of Q, say Q(X), is used as a measure of the representation capabilities of the prototype taken with respect to X. For the single-element clusters, card(X) = 1, the elements of the clusters are obviously ideal prototypes and the above expression always equals 1. In sequel, the global sum of Q taken over all the clusters "c",

$$V(c) = \sum_{\text{all clusters}} Q(\mathcal{X})$$
(10)

could be admitted as an indicator of representativeness of the data conveyed by their clusters (more precisely, their prototypes). For c = N one has V(c) = N. Generally speaking, V(c) is a non-decreasing function of the number of clusters, namely $V(c_1) \leq V(c_2)$ for $c_2 > c_1$. The analysis of the behaviour of V(c) being plotted versus c could be used to detect the most "plausible" number of clusters: the minimal value of c, say, c^*, that does not lead to a substantial and abrupt decrease in V can be accepted as a viable candidate for the structure in this set of patterns. Similarly, as envisioned in the case of cluster validity indices [1], V(c) should be viewed as a measure indicating a range (rather than a single specific number) of plausible clusters worth considering in the data set.

3. 2. Construction of local model induced by clusters

The data grouped in the i-th cluster are approximated in the obvious way by the corresponding local model

$$y = M_i(x, a_i)$$
(11)

i=1, 2, ..., c where a_i denotes a collection (vector) of the parameters of the model. As the hierarchical clustering is two-valued (the clusters are sets rather than fuzzy sets of data points), the identification of the corresponding local model embraces all these data points belonging to this particular cluster. As mentioned, the form of the model is not restricted in any way so finally the multimodel can include a variety of linear and nonlinear models, fuzzy relational equations, neural networks, fuzzy neural networks, etc. This means that in general the mapping capabilities of the multimodel can be made as high as required.

When using the multimodel, each local model becomes invoked with a certain confidence m_i depending on a position of x with respect to the prototype of this model p_{xi}. The pertinent calculations of the confidence level are realized by the matching module associated with this local model. More specifically,

$$\mu_i = \frac{1}{n} \sum_{k=1}^{n} (x_k \equiv p_{xi})$$
(12)

where $p_{xi} = [p_{1xi}\ p_{2xi} \dots p_{nxi}]$ and $x = [x_1\ x_2 \dots x_n]$.

4. Numerical experiments

The following examples including several collections of synthetic data serve as an illustration of the proposed method. In all these experiments, the objective function exploits the implication and equality index based on the implication given by (4). This choice is somewhat arbitrary; any other realization of the implication operator could lead to slightly different numerical results. For illustrative reasons the examples are two-dimensional.

Example 1. The data analyzed here, Fig.5, are distributed very regularly forming a simple geometrical construction. By eyballing the plot of the obtained objective function, Fig. 6, we can admit c=4 a equal to the minimal number of the clusters identified in the considered data set.

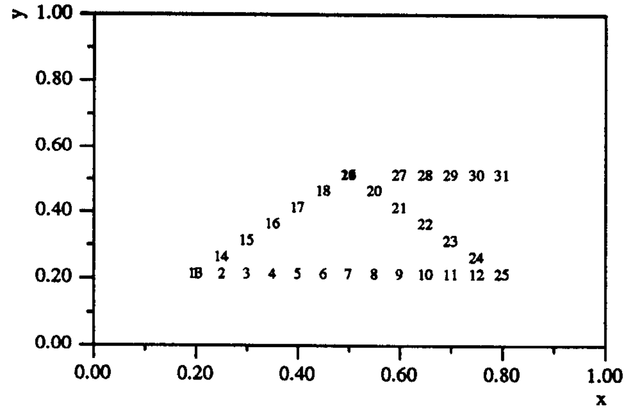

Fig. 5. Two - dimensional data set

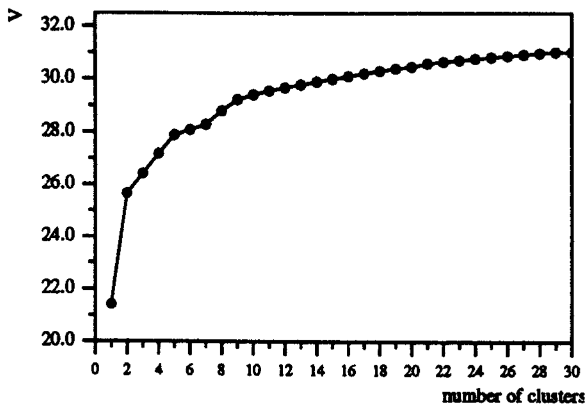

Fig. 6. V for several values of "c"

The prototypes of the clusters assume the values

$$\begin{bmatrix} 0.35 & 0.35 \\ 0.65 & 0.35 \\ 0.50 & 0.20 \\ 0.60 & 0.50 \end{bmatrix}$$

while the obtained partition of the data is illustrated in Fig.7(a).

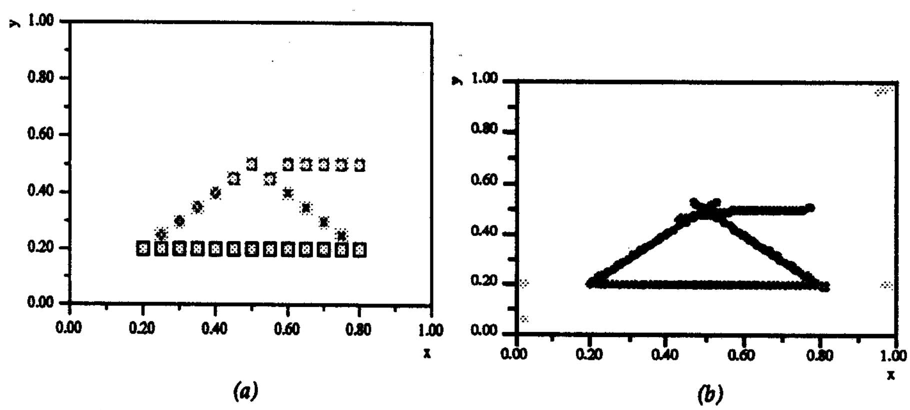

Fig. 7. Partition of ata (a) and fuzzy multimodel with the threshold level 0.8(b)

The models (M_1, M_2, M_3, M_4) are constructed locally within each cluster (all the models are linear except the one for M_4 described as $y = 1.204\,x^3 - 2.668\,x^2 + 1.99\,x$). The obtained models are outlined in Fig.1(b).

Example 2. We look more carefully at the s - curve characteristic for bifurcation diagrams illustrating relationship between temperature (T) and the Damk Öhler number (D), refer also to Fig. 1(d). These diagrams exhibit a multiplicity of steady states. We study a finite collection of data as shown in Fig. 8.

The directionality component plays a vital role in the revealing the relationships between the variables. Without it the clustering method was not capable of identifying any useful relationship even for a relatively high number of the clusters forced upon the data set. Below summarized are the partitions obtained when clustering the data without the use of the directionality component and with its inclusion.

Partition of data - no directionality component:

5 clusters:

{ 2 1}
{6 5 4 3}
{12 11 8 7 9 10}
{22 21 20 19 18 14 13 17 16 15}
{ 26 25 24 23}

4 clusters:

{6 5 4 3 2 1}
{12 11 8 7 9 10}
{22 21 20 19 18 14 13 17 16 15}
{26 25 24 23}

3 clusters:

{6 5 4 3 2 1}
{12 11 8 7 9 10}
{26 25 24 23 22 21 20 19 18 14 13 17 16 15}

Partition of data - directionality component included (3 clusters)
{10 9 8 7 6 5 4 3 2 1}
{14 13 12 11 17 16 15}
{26 25 24 23 22 21 20 19 18}

As expected the linear models were capable of reconstructing the data, Fig. 9

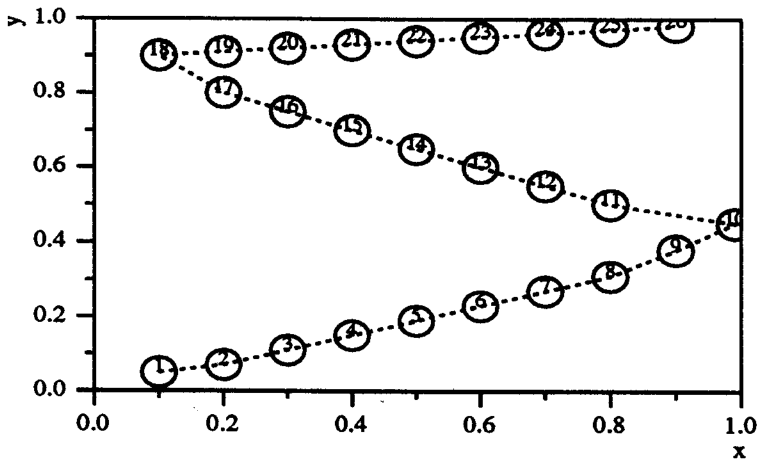

Fig. 8. s - curve - data set

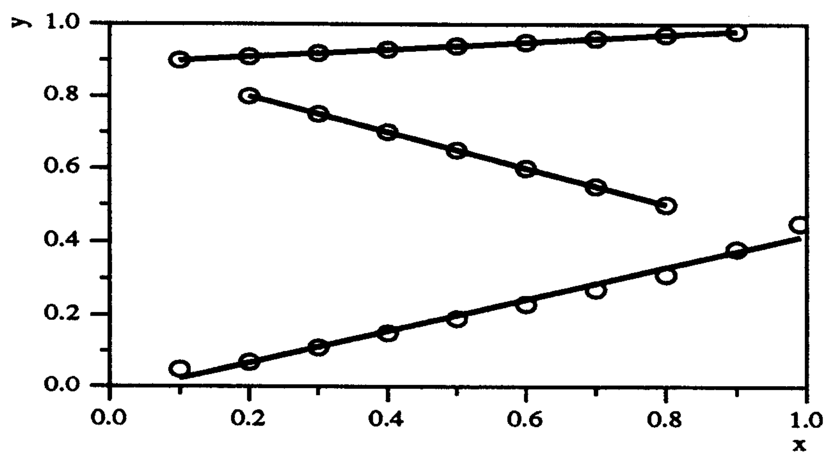

Fig. 9 Fuzzy multimodel composed of three models

5. Conclusions

The concept of fuzzy multimodelling opens up a new avenue of system modelling with fuzzy sets. Within this framework we treat the multiplicity of possible output values of the multimodel not as a deficiency of the modelling technique but rather as its genuine advantage. The multimodel places the idea of randomness in a novel context of structural varieties that are operationally captured in terms of fuzzy relations. While this study has primarily focussed on the fundamentals of fuzzy multimodelling, some essential design issues are still left open:

- firstly, the choice of the admission threshold g^* has to be investigated in depth along with some parametric adjustments (parameters a and b) of the objective function. This could eventually involve some scenarios of limited supervision involving samples of data with some indications about the admitted level of the structural variability. Similarly, more detailed guidelines should be established with respect to the number of the clusters (models) contributing to the multimodel.

-secondly, one can exploit various types of local models and experimentally quantify their computational features as well as pay a particular attention to generalization capabilities achieved by the entire multimodel.
These aspects will be pursued in future studies.

Acknowledgements

Support from the Natural Sciences and Engineering Research Council of Canada, MICRONET and the University Research Grants Program (URGP) of the University of Manitoba is gratefully acknowledged.

6. References

1. J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
2. J.C.Bezdek, S.K.Pal, Fuzzy Models for Pattern Recognition, IEEE Press, New York, 1992.
3. R. J. Hathaway, J. C. Bezdek, Switching regression models and fuzzy clustering, IEEE Trans. on Fuzzy Systems, 1, 1993, 195 - 204.
4. A. Kandel, Fuzzy Techniques in Pattern Recognition, J. Wiley, New York, 1982.
5. W. Pedrycz, 1984, Construction of fuzzy relational models, Cybernetics and Systems Research, Vol. 2, R. Trappl, ed., North Holland, Amsterdam, pp. 545–549.
6. W. Pedrycz, Direct and inverse problem in comparison of fuzzy data, Fuzzy Sets and Systems, 34, 1990, 223-236.
7. W. Pedrycz, Fuzzy Control and Fuzzy Systems, 2nd extended edition, Research Studies Press/J.Wiley, Taunton/New York, 1993.
8. W. Pedrycz, Fuzzy Sets Engineering, CRC Press, Boca Raton, FL, 1995.
9. M. Sugeno, T. Yasukawa, A fuzzy logic based approach to qualitative modeling, IEEE Trans. on Fuzzy Systems, 1, 1993, 7 - 31.
10. L. A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Trans. on Systems, Man, and Cybernetics, 3, 1973, 28 - 44.