A Fuzzy-neuro system for reconstruction of multi-sensor information

Simon Petit-Renaud and Thierry Denœux
Université de Technologie de Compiègne,
HeuDiaSyc-UMR C.N.R.S. 6599, BP 20529, 60205 Compiègne, France
phone: 33 (0)3 44 23 44 96, fax: 33 (0)3 44 23 44 77
e-mail: spetit, tdenoeux@hds.utc.fr

Abstract. In this paper, we propose a fuzzy-neuro method for reconstruction of missing data. This method allows to express knowledge acquired from input-output data in the form of fuzzy linguistic IF-THEN rules. The construction of these rules is based on the relations between the components of reference vectors in the measurement space. The method provides fuzzy estimates of all the missing components of a vector, given an arbitrary number of available variables. An example taken from an environmental monitoring application demonstrates the effectiveness of this approach.

Keywords: fuzzy rules, missing data reconstruction, training algorithm, neural networks

To appear in the proceedings of Fuzzy-Neuro Systems'98, March 19-20, 1998, Munich.

1 Introduction

In real applications, classical models from statistics, pattern recognition or control theory are not always well adapted to the actual knowledge of the studied system. For instance, some particularly informative parameters may not be available at a given time. In this paper, we propose a new technique for the *reconstruction* of missing values in a measurement vector. This approach is based on fuzzy models, which are known for their capability to express vague relations between the variables of a system [12,11]. This formalism allows to represent the uncertainty attached to estimates of missing variables in the form of a fuzzy sets. Moreover, unlike classical regression methods, our approach provides an estimation of *any* number of missing components in a vector, using all available ones.

In order to provide an analytical description of the relationship between inputs and outputs of a system, many methods have been proposed, combining fuzzy models and neural networks [8,10]. The representation of fuzzy models as neural networks allows to apply powerful learning techniques for identifying the structure of the model [1,2,9]. Neural fuzzy systems combine the advantages of both neural networks and fuzzy systems: they are universal approximators [4] and they can be trained by back propagation; their parameters and structure have strong physical meaning and *a priori* knowledge can be effectively incorporated in their initial parameter values. For these reasons, the fuzzy-neuro approach has been adopted in our model.

2 Fuzzy system models

One of the main directions in the theory of fuzzy systems is the linguistic approach initially introduced by Zadeh [12]. A Linguistic Model describes a system by means of a set of IF-THEN rules with vague predicates. The general model of a fuzzy system is based on a collection of linguistic rules between the variables:

$$\text{IF} \quad x_1 \text{ is } B_{i,1} \ldots \text{ and } \ldots x_r \text{ is } B_{i,r} \quad \text{THEN} \quad y \text{ is } D_i, \tag{1}$$

where $x_1, \ldots, x_r$ are the input variables of the base sets $X_1, \ldots, X_r$, and y is the output variable taking values in Y. $B_{i,j}$ and D_i are linguistic terms of the base sets X_j and Y, respectively. These fuzzy sets are characterized by their membership functions $\mu_{B_{i,j}}$, and μ_{D_i}. They constitute the parameters of the model. The number of rules determines the structure of the linguistic model. Two alternative reasoning mechanisms may be employed with such models. We will focus on the constructive method [7], in which the aggregation of the rules is performed as the union of fuzzy relations. There are also many different choices concerning the operators in the fuzzy inference system [6]. In order to have a differentiable expression between the inputs and the outputs of the system, we chose the *sum-product* rule of inference. Using this rule, the fuzzy output S of the system is given by the following expression: $\mu_S(y) = \sum_{i=1}^{m} \tau_i(\boldsymbol{x})\mu_{D_i}(y)$, where τ_i is the degree of firing of the i^{th} rule for the input $\boldsymbol{x} = (x_1, \ldots, x_r) \in X = X_1 \times \ldots \times X_r$.

In order to obtain a crisp output value, some *defuzzification* of the fuzzy output S must be performed. The usual *Center of Area* (COA) method provides the following expression for this crisp output:

$$y = f(\boldsymbol{x}) = \frac{\int_Y u\mu_S(u)du}{\int_Y \mu_S(u)du} = \frac{\sum_{i=1}^{m} y_i^\star \tau_i \int_Y \mu_{D_i}(u)du}{\sum_{i=1}^{m} \tau_i \int_Y \mu_{D_i}(u)du}, \tag{2}$$

where $f : X \longmapsto Y$ and $y_i^\star$ is the defuzzified value of the consequent fuzzy set D_i. Thus, the mapping f, determined by (2), can be represented as a three layer-feedforward network [5]. This method of simplified reasoning opens the possibility of learning fuzzy models from data.

3 Application to missing data reconstruction

In this paper, we use a set of fuzzy rules to solve a specific problem: missing data reconstruction. The principle of the method is the following.

Let $\mathcal{X}$ be a training set of N vectors of r variables that may have missing components, and $C = \{c_i \in \mathbb{R}^r, i \in \{1, \dots, m\}\}$, a set of m prototypes. The prototypes, which are supposed to summarize the information given by all the vectors, may have been created by means of any clustering method such as the c-means or fuzzy c-means algorithms (note that these methods can take incomplete vectors into account). Each vector $\boldsymbol{x} \in \mathcal{X}$ may be associated to the nearest prototype $c(\boldsymbol{x})$, according to a given distance: $c(\boldsymbol{x}) = \arg\min_{i \in C} \|\boldsymbol{x} - c_i\|$. For every $i \in \{1, \dots, m\}$, we can consider the set of vectors which are associated to the same vector c_i: $P_i = \{\boldsymbol{x} \in \mathcal{X}, c(\boldsymbol{x}) = c_i\}$. $\mathcal{X}$ is thus partitioned in m classes $(P_i)_{i=1}^m$.

If the prototypes provide a good representation of the data, we can hope to deduce the missing components of a vector close to a given prototype by means of the other components. Each prototype will then define a rule expressing the relationship between the different components of the corresponding class.

3.1 Construction of the fuzzy sets

As mentioned before, the essential elements of a fuzzy model are the linguistic terms in the antecedent and consequent parts of the rules. Here, we propose to associate a rule to each prototype or to each class. For every $i \in \{1, \dots, m\}$, let B_{ij} be a fuzzy set representing variable j of class P_i. $B_{i,j}$ can be defined by two parameters:

- c_{ij}, the j^{th} variable of the i^{th} prototype,
- σ_{ij}, the standard deviation of variable x_j in class P_i, $\sigma_{ij}^2 = \frac{1}{|Q_{ij}|} \sum_{\boldsymbol{x} \in Q_{ij}} (x_j - c_{ij})^2$ where $Q_{ij} \subset P_i$ represents the set of vectors of class P_i, for which parameter j is known.

The parameters of the model have a clear interpretation: roughly speaking, σ_{ij} measures the spread of variable j in class P_i, and the interval $[c_{ij} - \sigma_{ij}, c_{ij} + \sigma_{ij}]$ contains the "most possible" values. The shape of the membership function is more or less arbitrary. For practical reasons, we chose normalized Gaussian membership functions, defined by $\mu_{B_{ij}}(u) = \exp\left\{-\frac{1}{2}\left(\frac{u - c_{ij}}{\sigma_{ij}}\right)^2\right\}, \forall u \in \mathbb{R}$.

3.2 Missing data reconstruction

The problem can now be formulated as follows. Let $\boldsymbol{x} = (x_1, \dots, x_r) \in \mathbb{R}^r$. Let us suppose that there is a subset $I \subset \{1, \dots, r\}$ of components which are unknown or have been rejected as outliers. Let us note $V = \{1, \ldots, r\} \backslash I$. Then, for each $l \in I$ and for each prototype i, we obtain a rule $R_{i,l}$:

$$\text{IF } \bigwedge_{j \in V} x_j \text{ is } B_{ij} \text{ THEN } x_l \text{ is } B_{il} \tag{3}$$

Using the sum-product rule of inference, the degree of firing τ_i of rule R_{il} is defined as:

$$\tau_i = \prod_{j \in V} \mu_{B_{ij}}(x_j) = \exp\left\{ -\frac{1}{2} \sum_{j \in V} \left(\frac{x_j - c_{ij}}{\sigma_{ij}} \right)^2 \right\} \tag{4}$$

In the simplified reasoning method, the defuzzified outputs $\widehat{x}_l$ of the fuzzy system, which are estimates of the corresponding x_l, are easily obtained, according to (2), as

$$\widehat{x}_l = \frac{\sum_{i=1}^m \tau_i \sigma_{il} c_{il}}{\sum_{i=1}^m \tau_i \sigma_{il}} = \sum_{i=1}^m v_{il} c_{il} \quad \forall l \in I, \tag{5}$$

where c_{il} is the defuzzified value of the consequent part B_{il}, and $v_{il} = \tau_i \sigma_{il} / \sum_{k=1}^m \tau_k \sigma_{kl}$ is the normalized degree of firing. The estimated value of each missing component is thus a linear combination of the corresponding components of the prototypes. The weight v_i reflects the influence of rule i in term of distance between the considered prototype and the reconstructed pattern.

Note that, by (4) and (5), our model can be viewed as an auto-associative radial basis function neural network. At this step, we can exploit the classical advantage of neural networks: the possibility of identifying the model from training data.

4 Learning parameters by back-propagation

Generally, the identification of a fuzzy system model consists in two major phases: structure identification, and parameter learning. Structure identification includes the choice of input and output variables, and the determination of the form and number of rules. In general, structure identification is extremely difficult. However, in our case, the problem becomes simply to determine the number of rules m, which can be done very easily using, e.g., a cross-validation procedure. We therefore concentrate in this paper on parameter estimation, which consists here in the determination of the σ_{ij} and c_{ij}.

By (4) and (5), we have already seen that the system can be represented as an associative three-layer network. Thus, for a fixed number of rules m, the problem of setting up the fuzzy model becomes essentially a learning problem, which may be formalized as the minimization of an empirical error function, such as:

$$e = \frac{1}{2|I|} \sum_{l \in I} (\widehat{x}_l - x_l)^2, \tag{6}$$

with respect to σ_{ij} and c_{ij}. In order to tune these parameters, the following adaptive descent method may be used. First, the initial values $c_{ij}(0)$ and $\sigma_{ij}(0)$ are chosen using the partition of the training set, as explained in Section 3.1. At each iteration t, we then compute:

$$c_{ij}(t+1) = c_{ij}(t) - \eta(t)\frac{\partial e}{\partial c_{ij}} \tag{7}$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \eta(t)\frac{\partial e}{\partial \sigma_{ij}}, \tag{8}$$

where $\eta(t)$ is an adaptive step depending on the iteration. The error derivatives are easily obtained by straightforward calculus as:

$$\frac{\partial e}{\partial c_{ij}} = \frac{x_j - c_{ij}}{(\sigma_{ij})^2} \sum_{l \in V} (\widehat{x}_l - x_l)(c_{il} - \widehat{x}_l)v_{il} + v_{il}(\widehat{x_j} - x_j), \tag{9}$$

$$\frac{\partial e}{\partial \sigma_{ij}} = \frac{(x_j - c_{ij})^2}{(\sigma_{ij})^3} \sum_{l \in V} (\widehat{x}_l - x_l)(c_{il} - \widehat{x}_l)v_{il} + \frac{v_{ij}}{\sigma_{ij}}(c_{ij} - \widehat{x_j}). \tag{10}$$

The training algorithm is based on the error back-propagation procedure. It works in two passes. In the forward pass, one calculates the current degree of firing of the rules, their normalized values and the estimated crisp outputs of the system. Current estimates $c_{ij}(t)$ and $\sigma_{ij}(t)$ of c_{ij} and σ_{ij} are used in this calculation. In the backward pass, the current parameter estimates $c_{ij}(t)$ and $\sigma_{ij}(t)$ are updated according the learning rules. The algorithm is iterated until some convergence criterion is met.

5 Fuzzy confidence interval

When some unknown component in an input vector is estimated, there is inevitably some uncertainty attached to the estimate, which is due to the imperfectness of both the completion procedure and the available information. In our approach, this uncertainty may be accounted for by providing fuzzy estimates of the unknown quantities. For that purpose, we need to use the whole fuzzy output and not only the defuzzified value, in order to provide a complete possibility distribution. Dubois and Prade [3] generalized the notion of interval in the fuzzy sets theory, by introducing the concepts of fuzzy number and fuzzy interval. These notions require some regular properties of the fuzzy sets, particularly quasiconcavity of the membership function.

In general, the fuzzy outputs $(S_l)_{l \in I}$ of our system are neither fuzzy numbers, nor fuzzy intervals. The membership function of S_l has the following expression:

$$\mu_{S_l}(u) = \sum_{i=1}^{m} \tau_i \mu_{B_i}(u), \qquad \forall u \in \mathbb{R}.$$

This membership function is a Gaussian mixture and is therefore not quasiconcave. However, fuzzy sets B_{ij} in our system are Gaussian LR fuzzy numbers, which may be denoted by: $\mu_{B_{ij}} = (c_{ij}, \sigma_{ij}, \sigma_{ij})_{L,R}$, with $L(u) = R(u) = \exp(-u^2)$ [3]. As the crisp estimate of x_l is $\widehat{x_l} = \sum_{i=1}^{m} v_{il} c_{il}$, it can be natural to construct the following output N_l:

$$\mu_{N_l} = \bigoplus_{i=1}^{m} v_{il} \mu_{B_{il}},$$

where $\bigoplus$ is the operator defining the sum of fuzzy numbers [3]. Using known properties concerning the addition of LR fuzzy numbers, we obtain:

$$\mu_{N_l} = \left(\sum_{i=1}^{m} v_{il} c_{il}, \sum_{i=1}^{m} v_{il} \sigma_{il}, \sum_{i=1}^{m} v_{il} \sigma_{il} \right)_{L,R}.$$

Consequently, N_l is also a Gaussian fuzzy number, whose standard deviation represents the imprecision concerning the estimated value. This choice is motivated by the simplicity and regularity of the final membership function. The defuzzified value deduced from μ_{N_l} is still $\widehat{x_l}$. This method allows to define confidence intervals for x_l by considering the α-cuts of N_l. These values may then be used as imprecise inputs to a decision system. However, uncertainty is not always well represented with this method, since the confidence of the crisp output $\widehat{x_l}$ is always equal to one. In case of conflicts, we may hesitate between different values corresponding to different local maxima of the fuzzy output S_l and the defuzzified value may simply result from a compromise between these values. This is the reason why S_l may be regarded as more informative than N_l.

6 Experiments

In order to demonstrate the validity of the proposed method, we considered real data from an environmental monitoring application, in which on-line measurements of four parameters (temperature, pH, conductivity and dissolved oxygen) were taken on-line in a river. A set of 700 examples was used for training, and the results were assessed using an independent test set of 700 examples. The number of rules was optimized by cross-validation, yielding $m = 10$. Therefore, we had $2mr = 80$ parameters to optimize. For a given vector to reconstruct, there are consequently $r(2^{r-1} - 1) = 28$ different possible models, since for each vector $\boldsymbol{x}$, each of the r variables $\boldsymbol{x}_j$ is reconstructed using one, two or three other variables. Table 1 compares the reconstruction error obtained with our method, according these kinds of models, with a simple reference method based on the replacement of missing components with the corresponding component of the nearest of the 10 prototypes. Figure 1 compares the non fuzzy outputs of the system for 200 samples, after 10 iterations in the optimisation phase, with the results from the same reference method. Figure 1 also shows an example of the fuzzy outputs of the system. The non-convexity of S justifies the use of fuzzy numbers N.

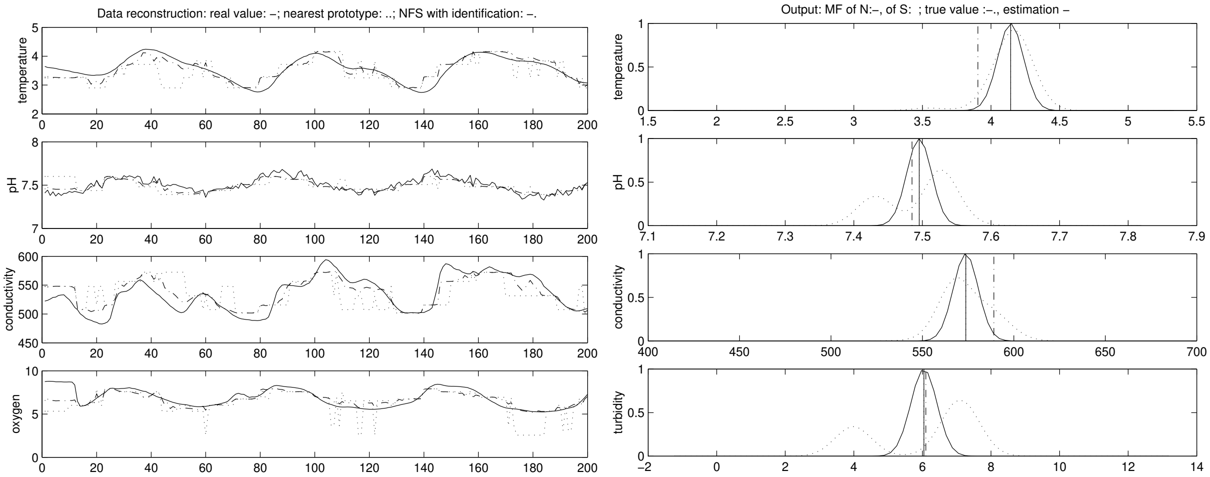

Figure 1: Data reconstruction: Left: real data (-); NF method with optimisation (t=10)(–); nearest prototype method (..); right: example of fuzzy ouputs for a given vector: fuzzy number N (-), fuzzy output S (..); real value (-.-); defuzzified value (-).

As shown from these figures and results, prior knowledge is efficiently incorporated in the system, and the results are significantly better than those obtained with the nearest prototype method. However, it can be noticed that some variables are not well reconstructed, which is probably due insufficient correlation with the other variables. Future works is expected to improve these first results.

Table 1: Reconstruction errors for different numbers of available input variables.

Mean error	1 input (12 models)	2 inputs (12 models)	3 inputs (4 models)
Fuzzy-Neuro Reconstruction	0.36	0.31	0.25
Nearest Prototype Reconstruction	0.51	0.46	0.42

7 Conclusion

A fuzzy-neuro system for missing data reconstruction has been described. This method expresses the knowledge acquired from input-output data in the form of fuzzy rules, and yields fuzzy estimates of all the missing components of a vector, given any number of available variables. The imprecision of the result can be represented in two forms, one of which is more informative but more complex, whereas the other allows to compute confidence intervals for the estimated values.

Acknowledgments

The authors are indebted to Stéphane Deveughèle for helpful comments on an early version of this paper. This work was partially supported by EEC funded Esprit Project 22442 (Environmental Monitoring and Management Systems).

Literature

[1] D. H. Bossley. *Neurofuzzy modelling approaches in system identification.* PhD thesis, University of Southampton, 1997.

[2] M. Brown and C. Harris. *Neurofuzzy adaptive modelling and control.* Prentice Hall, Hemel-Hampstead, 1994.

[3] D. Dubois and H. Prade. Additions of interactive fuzzy numbers. *IEEE Transactions on Automatic Control*, 26:926–936, 1981.

[4] K Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universel approximators. *Neural networks*, 2:359–366, 1989.

[5] J. S.R. Jang and C. T. Sun. Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Transactions on Neural Networks*, 4:156–159, 1993.

[6] C.C. Lee. Fuzzy logic in control systems: fuzzy logic controller, part 2. *IEEE Transactions on Systems, Man and Cybernetics*, 20:419–435, 1990.

[7] E. H. Mamdani. Twenty years of fuzzy control: Experience gained and lessons learnt. In *Proceedings 2nd International IEEE Conference on Fuzzy Systems*, pages 339–344, San Francisco, 1993.

[8] H. Nomura, I. Hayashi, and N. Wakami. A learning method of fuzzy inference rules by descent method. In *1st IEEE International Conference on Fuzzy Systems*, pages 203–210, San Diego, 1992.

[9] L. X. Wang. *Adaptive fuzzy system and control: design and stability analysis.* Prentice Hall, Englewood Cliffs, 1994.

[10] L. X. Wang and J. M. Mendel. Back-propagation fuzzy system as nonlinear dynamic system identifiers. In *1st IEEE International Conference on Fuzzy Systems*, pages 1409–1416, San Diego, 1992.

[11] R. R. Yager and D. P. Filev. *Essential of fuzzy modeling and control.* Wiley, 1994.

[12] L. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics*, 3:28–44, 1973.