amcs

ON CLASSIFICATION WITH MISSING DATA USING ROUGH–NEURO–FUZZY SYSTEMS

ROBERT K. NOWICKI

Institute of Information Technology
Academy of Management (SWSPiZ), ul. Sienkiewicza 9, 90–113 Łódź, Poland

Department of Computer Engineering
Częstochowa University of Technology, ul. Armii Krajowej 36, 42–200 Częstochowa, Poland
e-mail: rnowicki@kik.pcz.czest.pl

The paper presents a new approach to fuzzy classification in the case of missing data. Rough-fuzzy sets are incorporated into logical type neuro-fuzzy structures and a rough-neuro-fuzzy classifier is derived. Theorems which allow determining the structure of the rough-neuro-fuzzy classifier are given. Several experiments illustrating the performance of the rough-neuro-fuzzy classifier working in the case of missing features are described.

Keywords: fuzzy sets, rough sets, neuro-fuzzy architectures, classification, missing data.

1. Introduction

The classic fuzzy system depicted in (Lin and Lee, 1991; Lee and Kwang, 1994; Nauck *et al.*, 1997) as well as later ones, e.g., (Nowicki, 2000; Czogała and Łęski, 2000; Rutkowska and Nowicki, 2000a) are designed to work with some strictly determined set of input values. In the case of classification tasks, the decision is made based on known values of classified object features represented by a vector $\mathbf{v} = [v_1, v_2, \ldots, v_n]$. The values of features are denoted as the vector $\overline{\mathbf{v}} = [\overline{v}_1, \overline{v}_2, \ldots, \overline{v}_n]$. The goal of classification is to determine whether or not the object or state x belongs to class ω_j, $j = 1, \ldots, m$. Thus the classifier makes the decision if object x belongs to class ω_j ($x \in \omega_j$) or not ($x \notin \omega_j$). The assumption that the values of all n features are known is not always fulfilled. This occurs, e.g., in the case of medical or industrial diagnosis. In the first case, the lack of information is due to the impossibility of performing some tests with a patient in bad condition or when the test is unacceptable (due to an invasive method, cost or his/her faith). Moreover, some tests can be unnecessary when a classifier could make a certain decision without these test results. In industrial diagnosis, the classifier can work in on-line process monitoring and the process should still works even when some measurements are missing. Moreover, some values can be unnecessary when a classifier could make a certain decision

without these values.

Thus, we consider a set of n selected features, denoted as Q, which we use to describe objects (note that the number of real object features is unlimited) and to design the classifiers. When the systems are working, we have the information about the values of $n_D \leq n$ features. The set of features with known values is denoted by D, and the set of features with unknown values is denoted by G. Any information about features from outside the set Q has no significance—it cannot be used by the classifiers. Thus we can write down that $D \subseteq Q$ and $G = Q \setminus D$. The number of unknown values is $n_G = n - n_D$. In this paper we consider such a case.

Generally, in the case of classification with incomplete data, three approaches are well known in the literature:

- imputation,

- marginalisation,

- rough sets.

In the first approach, unknown values are replaced by estimated ones (Chan *et al.*, 1976; Dixon, 1979; Renz *et al.*, 2002; Tanaka *et al.*, 1996). The estimated value can be determined by any method, e.g., as the mean of known values of the same feature in other instances, by

a neural network, or using the nearest neighbour algorithm (Morin and Raeside, 1981). In the second approach, features with unknown values are ignored (Cooke *et al.*, 2001; Little and Rubin, 2002) and the problem boils down to classification in a lower-dimensional feature space. Here we can use any classifier, such as neural network (Bishop, 1995; Żurada, 1992), fuzzy systems (Lin and Lee, 1991; Wang, 1994; Zadeh, 1965; 1975), a nearest neighbor algorithm, statistical methods (Bishop, 1995; Duda *et al.*, 2001), the genetic approach (Fogel, 1995; Goldberg, 1989; Michalewicz, 1992), granular computing (Pedrycz and Bargiela, 2002; Yao and Yao, 2002), and support vector machines (Kecman, 2001; Surges, 1998). The third approach relies on rough set theory (Pawlak, 1982; 1991; 2002), which is a concept of description of the uncertainty of classified object taking into consideration limited knowledge about the object.

Imputation and marginalisation always lead to an increase in the number of mistakes when the number of unknown values increases. When we use rough set theory, the object can be classified into the positive region of the class (i.e., the object certainly belongs to the class), to the negative region of the class (i.e., the object certainly does not belong to the class), or to the boundary region of the class (i.e., it is not possible to determine if the object belongs to the class or not). The membership to the regions depends on the quality of object description. If the description is good enough, the object belongs to the positive or the negative region. If the description is too weak, then the object belongs to the boundary region.

In this paper we consider a fuzzy classifier which, due to cooperation with rough set theory, will give an answer only if the object description is good enough. Otherwise, it refrains from giving an answer. The classifier will apply well-known fuzzy rules (see Section 2) and, when all n values are known, will work exactly as its fuzzy progenitor. Moreover, we focus only on the logical type of fuzzy reasoning (Driankov *et al.*, 1993; Rutkowska and Nowicki, 2000a; Yager and Filev, 1994; Czogała and Łęski, 2000). This type of fuzzy system is more appropriate for classification tasks than Mamdani type or TSK systems, which was shown in (Nowicki and Rutkowska, 2000; Rutkowska and Nowicki, 2000a) and confirmed later in (Rutkowski and Cpałka, 2003; 2005) using flexible neuro-fuzzy systems. The learning (rule developing) phase is out of the scope of this paper. We assume that the appropriate fuzzy classifier has been built using any method (Fogel, 1995; Goldberg, 1989; Lin and Lee, 1991; Wang, 1994; Zadeh, 1965; 1975), and it performs satisfactorily when the values of all n input features (from the set Q) are available.

A similar approach based on Mamdani type reasoning and CA (*center average*) defuzzification as well as MICOG (*modified indexed centre of gravity*) defuzzifications (both Mamdani and logical type of reasoning) were studied in (Nowicki, 2008; 2009), respectively.

The main contribution of this paper is the development of original architectures of rough-neuro-fuzzy classifiers based on logical reasoning and DCOG defuzzification. Experiments illustrate the performance of rough-neuro-fuzzy classifiers working in the case of missing features.

In the paper we first study the neuro-fuzzy structures (NFSs) based on genuine (logical) inference (Section 2). Next (Section 3), the rough fuzzy set definition and some aspects of it are presented. Then (Section 4), logical neuro-fuzzy classifiers based on DCOG defuzzification are defined. Finally, logical neuro-fuzzy classifiers are converted into the logical rough-neuro-fuzzy classifier (RNFC), and the correctness of this conversion is proven.

We would like to emphasise here that rough-neuro-fuzzy classifiers proposed in this paper do not give better results compared with the corresponding neuro-fuzzy classifiers. When all features are available, the performance of both classifiers is the same. However, a great advantage of rough-neuro-fuzzy classifiers lies in unambiguously defined classification in the case of missing features (see Theorem 1 in Section 5 and experiments with discussions in Section 6).

2. NFS elements

2.1. Fuzzy inference. Fuzzy inference systems realise fuzzy reasoning founded on fuzzy rule bases. When we assume that $\mathbf{v} = [v_1, v_2, \ldots, v_n]$ is a vector of features describing any object or state and $\mathbf{y} = [y_1, y_2, \ldots, y_m]$ is the vector of the output values of a system, the rules are represented in the form

$$
\begin{aligned}
R^r : \text{ IF } & v_1 \text{ is } A_1^r \text{ AND } v_2 \text{ is } A_2^r \\
& \text{AND} \ldots \text{ AND } v_n \text{ is } A_n^r \text{ THEN } y_1 \text{ is } B_1^r, \\
& \qquad y_2 \text{ is } B_2^r, \ldots, y_m \text{ is } B_m^r, \quad (1)
\end{aligned}
$$

where $\mathbf{v} \in \mathbf{V} = \mathbf{V}_1 \times \mathbf{V}_2 \times \ldots \times \mathbf{V}_n, \mathbf{y} \in \mathbf{Y} = \mathbf{Y}_1 \times \mathbf{Y}_2 \times \ldots \times \mathbf{Y}_m$ and $A^r = A_1^r \times A_2^r \times \ldots \times A_n^r \subseteq \mathbf{V}$, $B^r = B_1^r \times B_2^r \times \ldots \times B_m^r \subseteq \mathbf{Y}$ are fuzzy sets.

In the literature various neuro-fuzzy systems have been proposed (Lee and Kwang, 1994; Lin and Lee, 1991; Nauck *et al.*, 1997; Nowicki, 2004; Rutkowska and Nowicki, 2000b; Rutkowski and Cpałka, 2005).

Genuine fuzzy implications are used in the logical approach (Czogała and Łęski, 2000; Rutkowska and Nowicki, 2000a; Rutkowska *et al.*, 2000). We can enumerate some groups of genuine fuzzy implication (Mas *et al.*, 2007):

- S-implications:

$$
I(a, b) = S\{N\{a\}, b\}. \quad (2)
$$

The Łukasiewicz, Reichenbach, Kleene-Dienes, Fodor and Dubois-Prade implications are examples of S-implications.

- R-implications:

$$I(a,b) = \sup_{z \in [0,1]} \{z | T\{a,z\} \le b\}. \qquad (3)$$

The Rescher, Goguen and Gödel implications are examples of R-implications.

- QL-implications:

$$I(a,b) = S\{N\{a\}, T\{a,b\}\}. \qquad (4)$$

The Zadeh implication is an example of QL-implications.

- D-implications:

$$I(a,b) = S\{T\{N\{a\}, N\{b\}\}, b\}. \qquad (5)$$

In (2)–(5), $a, b \in [0,1]$, T is any t-norm, S is any t-conorm, N is any fuzzy negation (Klement *et al.*, 2000). It should be noted that S-implications and R-implications fulfil the fuzzy implication definition proposed in (Fodor, 1991).

2.2. Defuzzification. One of the most important elements (besides implication) determining the architecture of such systems is defuzzification. In the sequel we shortly review and discuss defuzzification methods used in designing neuro-fuzzy systems. By T and S we denote the t-norm and the t-conorm, respectively.

As a fundamental method of defuzzification we can find the *centre of gravity defuzzification* (COG), also called the *centre of area defuzzification* (COA), defined by

$$\overline{y}_j = \frac{\int\limits_{y_j \in \mathbf{Y}_j} y_j \cdot \mu_{B'_j}(y_j)\, \mathrm{d}y_j}{\int\limits_{y_j \in \mathbf{Y}_j} \mu_{B'_j}(y_j)\, \mathrm{d}y_j}, \qquad (6)$$

where B'_j, $j = 1, \ldots, m$, is an aggregated conclusion of reasoning for all rules. The membership function $\mu_{B'_j}(y_j)$ is calculated in the logical approach of fuzzy reasoning as follows:

$$\mu_{B'_j}(y_j) = \mathop{T}\limits_{r=1}^{N} \mu_{B''^r_j}(y_j), \qquad (7)$$

where

$$\mu_{B''^r_j}(y_j) = I(\tau^r, \mu_{B^r_j}(y_j)), \qquad (8)$$

with I being a fuzzy implication (see Section 2), τ^r the so-called "rule activation level" or "level of rule firing" defined as

$$\tau^r = \mu_{A^r}(\overline{\mathbf{x}}), \qquad (9)$$

and $\overline{\mathbf{x}} = [\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n]$—the vector of the values of features **x**.

A problem with COG defuzzification occurs when the integrals in (6) approach infinity. This happens in many fuzzy implications, e.g., S-implications and QL-implications. Moreover, in real implementation, the integrals in (6) should be discretized or approximated (Patel and Mohan, 2002) in order to derive a neuro-fuzzy system. Certain simplifications of COG defuzzification derivation are available due to limitation to the specific shape of fuzzy sets applied (Broekhoven and Beats, 2006).

The first propositions of neuro-fuzzy systems (Wang, 1994) used *centre average defuzzification* (CA) defined by

$$\overline{y}_j = \frac{\sum\limits_{r=1}^{N} \overline{y}^r_j \cdot \mu_{B''^r_j}(\overline{y}^r_j)}{\sum\limits_{r=1}^{N} \mu_{B''^r_j}(\overline{y}^r_j)}, \qquad (10)$$

where

$$\mu_{B''^r_j}(\overline{y}_j) = T(\mu_{A^r}(\overline{\mathbf{x}}), \mu_{B^r_j}(\overline{y}_j)). \qquad (11)$$

Regrettably, this method is improper in the case of the fuzzy reasoning based on the genuine fuzzy implications.

The drawback of the CA defuzzification method, given by (10), is that it is useful only in the case of the Mamdani approach (11) to fuzzy reasoning. It is easily seen that when dealing with the logical approach we have $h(B''^r_j) = 1$. This drawback is removed if we apply the *discrete centre of gravity defuzzification* (DCOG) proposed and first used in (Nowicki, 2000; Rutkowska and Nowicki, 2000a; Rutkowska *et al.*, 2000). It is defined by

$$\overline{y}_j = \frac{\sum\limits_{r=1}^{N} \overline{y}^r_j \cdot \mu_{B'_j}(\overline{y}^r_j)}{\sum\limits_{r=1}^{N} \mu_{B'_j}(\overline{y}^r_j)}, \qquad (12)$$

where B'_j is calculated as in the COG method. Note that also in this method (as in CA) the shape of the membership functions is not precisely taken into account.

Another solution appropriate for genuine fuzzy implications is MICOG (*modified indexed centre of gravity*) defuzzification discussed in (Czogała and Łęski, 2000) and used for the case of missing features in (Nowicki, 2008).

3. Rough fuzzy sets

The concept of using rough sets and fuzzy sets together comes from Dubois and Prade (Dubois and Prade, 1990; 1992). They proposed two approaches to combining both theories. The first one leads to the definition of the rough fuzzy set, where lower and upper approximations of a fuzzy set are defined. The second one leads to the (more general) definition of the fuzzy rough set, where the lower and upper approximations of a fuzzy set are also fuzzy.

The rough fuzzy set is a pair $\left(\underline{R}A, \overline{R}A\right)$ of fuzzy sets. $\underline{R}A$ is an R-lower approximation and $\overline{R}A$ is an R-upper approximation of fuzzy set $A \subseteq X$. The membership functions of $\underline{R}A$ and $\overline{R}A$ are defined as follows:

$$\mu_{\underline{R}A}(\hat{x}) = \inf_{x \in [\hat{x}]_R} \mu_A(x), \qquad (13)$$

$$\mu_{\overline{R}A}(\hat{x}) = \sup_{x \in [\hat{x}]_R} \mu_A(x), \qquad (14)$$

where $[\hat{x}]_R$ is an equivalence class (Polkowski, 2002).

An object $x \in \mathbf{X}$ is described by a vector of features $\mathbf{v} \in \mathbf{V}$, so let us equate its membership with the membership of its feature values $\overline{\mathbf{v}} = [\overline{v}_1, \overline{v}_2, \ldots, \overline{v}_n]$. Consequently, we can use x or $\overline{\mathbf{v}}$ interchangeably. Let us assume that the fuzzy set $A \subseteq \mathbf{V}$ is given along with its membership function $\mu_A(x) = \mu_A(\overline{\mathbf{v}}) = \mu_A(\overline{v}_1, \overline{v}_2, \ldots, \overline{v}_n)$, where $\overline{v}_i \in \mathbf{V}_i = \mathbb{R}$ for $i = 1, \ldots, n$ and $\mathbf{V} = \mathbb{R}^n$. We also define the set of all features of object x as $Q = \{v_1, v_2, \ldots, v_n\}$. Let us isolate the subset of features $D \subseteq Q$. The $\widetilde{D}$-indiscernibility relation is defined by

$$x \widetilde{D} \hat{x} \Leftrightarrow \forall v \in D; f_x(v) = f_{\hat{x}}(v), \qquad (15)$$

where $x, \hat{x} \in \mathbf{X}$ and f_x is an information function expressing the value of feature v_i of object x. The membership functions of lower and upper approximations of the rough fuzzy set $\widetilde{D}A$ can be described by

$$\mu_{\underline{\widetilde{D}}A}(x) = \inf_{\overline{\mathbf{v}}_G \in \mathbf{V}_{Q \setminus D}} \mu_A(\overline{\mathbf{v}}_D, \overline{\mathbf{v}}_G), \qquad (16)$$

and

$$\mu_{\overline{\widetilde{D}}A}(x) = \sup_{\overline{\mathbf{v}}_G \in \mathbf{V}_{Q \setminus D}} \mu_A(\overline{\mathbf{v}}_D, \overline{\mathbf{v}}_G). \qquad (17)$$

If we assume that the fuzzy set A^r, $r = 1, \ldots, N$ is a Cartesian product, i.e., $A^r = A_1^r \times A_2^r \times \ldots \times A_n^r$, then the membership function of its $\widetilde{D}$-lower approximation is given by the following equation:

$$\mu_{\underline{\widetilde{D}}A^r}(x) = T \left\{ \underset{i: v_i \in D}{T} \mu_{A_i^r}(\overline{v}_i), \underset{i: v_i \in G}{T} \inf_{\overline{v}_i \in \mathbf{V}_i} \mu_{A_i^r}(\overline{v}_i) \right\}. \qquad (18)$$

The membership function of its $\widetilde{D}$-upper approximation is given by the equation

$$\mu_{\overline{\widetilde{D}}A^r}(x) = T \left\{ \underset{i: v_i \in D}{T} \mu_{A_i^r}(\overline{v}_i), \underset{i: v_i \in G}{T} \sup_{\overline{v}_i \in \mathbf{V}_i} \mu_{A_i^r}(\overline{v}_i) \right\}. \qquad (19)$$

4. Logical type neuro-fuzzy classifier

The rules depicted in (1) are suitable for approximation and majority of control tasks. In the case of classification tasks, rules in other forms are more appropriate. Generally, when the membership of object x to class ω_j specified in the consequent of rules is fuzzy ($\overline{z}_j^r = \mu_{\omega_j}(x)$),

in the case of independent variables v_i the rules have the following form (Kuncheva, 2000):

R^r: IF v_1 is A_1^r AND v_2 is A_2^r
   AND ... AND v_n is A_n^r THEN $x \in \omega_1(\overline{z}_1^r)$,
   $x \in \omega_2(\overline{z}_2^r), \ldots, x \in \omega_m(\overline{z}_m^r)$,   (20)

where $r = 1, \ldots, N$, N is the number of rules and $\overline{z}_j^r$ is the membership degree of the object x to the j-th class ω_j according to rule r.

Let us assume that the membership of objects to classes is not fuzzy but crisp, i.e.,

$$\overline{z}_j^r = \begin{cases} 1 & \text{if } x \in \omega_j, \\ 0 & \text{if } x \notin \omega_j. \end{cases} \qquad (21)$$

We write just $x \in \omega_j$ when $\overline{z}_j^r = 1$ (which means that object x belongs to the j-th class, according to the r-th rule) in the definition of the r-th rule. We can omit the part $x \in \omega_j(\overline{z}_j^r)$ when $\overline{z}_j^r = 0$ (which means that object x does not belong to the j-th class, according to the r-th rule).

It is easy to notice that variables $\overline{z}_j^r$ correspond to variables $\overline{y}_j^r$ in Section 2 subject to the assumption (21). Thus we can use rules presented in the description (1) in a specific form:

R^r: IF v_1 is A_1^r AND v_2 is A_2^r
   AND ... AND v_n is A_n^r THEN z_1 is B_1^r,
   z_2 is $B_2^r, \ldots, z_m$ is B_m^r.   (22)

Fuzzy sets $B_1^r, B_2^r, \ldots, B_m^r$ appearing in the antecedent of rules satisfy the equation

$$\mu_{B_j^r}(z_j) = \begin{cases} 1 & \text{if } z_j = \overline{z}_j^r, \\ 0 & \text{if } z_j = 1 - \overline{z}_j^r, \end{cases} \qquad (23)$$

where $\overline{z}_j^r$ fulfils the assumption (21). In special cases, fuzzy sets $B_1^r, B_2^r, \ldots, B_m^r$ could be a fuzzy singleton:

$$\mu_{B_j^r}(z_j) = \begin{cases} 1 & \text{if } z_j = \overline{z}_j^r, \\ 0 & \text{if } z_j \neq \overline{z}_j^r, \end{cases} \qquad (24)$$

but on account of the defuzzification method described by Eqn. (12) the value of $\mu_{B_j^r}(z_j)$ for $z_j \neq 0$ and $z_j \neq 1$ is without significance.

Using the above assumptions, we can adopt DCOG defuzzification as follows:

$$\overline{z}_j = \frac{\sum_{\substack{r=1 \\ r: \overline{z}_j^r = 1}}^{N} \mu_{B_j'}(\overline{z}_j^r)}{\sum_{r=1}^{N} \mu_{B_j'}(\overline{z}_j^r)}, \qquad (25)$$

and aggregation realised by any t-norm:

$$\mu_{B'_j}(z_j) = \overset{N}{\underset{r=1}{T}} \mu_{B''^r_j}(z_j). \tag{26}$$

The fuzzy set B''^r_j occurring in partial conclusions depends on implication:

- S-implications:

$$\mu_{B''^r_j}(z_j) = \begin{cases} 1 & \text{if } z_j = \overline{z}^r_j, \\ N\{\tau^r\} & \text{if } z_j \neq \overline{z}^r_j. \end{cases} \tag{27}$$

- R-implications:

$$\mu_{B''^r_j}(z_j) = \begin{cases} 0 & \text{if } z_j \neq \overline{z}^r_j \text{ and } \tau^r > 0, \\ 1 & \text{otherwise}. \end{cases} \tag{28}$$

Let us note that the case of R-implication is a special case of S-implication, because the expression

$$\begin{cases} 0 & \text{if } \tau^r > 0, \\ 1 & \text{otherwise} \end{cases}$$

is a special case of $N\{\tau^r\}$.

- QL-implications:

$$\mu_{B''^r_j}(z_j) = \begin{cases} S\{N\{\tau^r\}, \tau^r\} & \text{if } z_j = \overline{z}^r_j, \\ N\{\tau^r\} & \text{if } z_j \neq \overline{z}^r_j. \end{cases} \tag{29}$$

- D-implications:

$$\mu_{B''^r_j}(z_j) = \begin{cases} 1 & \text{if } z_j = \overline{z}^r_j, \\ N\{\tau^r\} & \text{if } z_j \neq \overline{z}^r_j. \end{cases} \tag{30}$$

So, we obtain descriptions of individual architectures of a neuro-fuzzy classifier:

- S-implications and D-implications:

$$\overline{z}_j = \frac{\displaystyle\sum_{\substack{r=1 \\ r:\overline{z}^r_j=1}}^{N} \overset{N}{\underset{\substack{k=1 \\ k:\overline{z}^k_j=0}}{T}} N\{\tau^k\}}{\displaystyle\sum_{r=1}^{N} \overset{N}{\underset{\substack{k=1 \\ k:\overline{z}^k_j \neq \overline{z}^r_j}}{T}} N\{\tau^k\}}, \tag{31}$$

- R-implications:
  The architecture is the same as described by Eqn. (31), for the negation defined as follows:

$$N\{\tau^r\} = \begin{cases} 0 & \text{if } \tau^r > 0, \\ 1 & \text{otherwise}. \end{cases} \tag{32}$$

- QL-implications:

$$\overline{z}_j = \frac{\displaystyle\sum_{\substack{r=1 \\ r:\overline{z}^r_j=1}}^{N} T\left\{ \begin{array}{l} \overset{N}{\underset{\substack{k=1 \\ k:\overline{z}^k_j=0}}{T}} N\{\tau^k\}, \\ \overset{N}{\underset{\substack{k=1 \\ k:\overline{z}^k_j=1}}{T}} S\{N\{\tau^k\}, \tau^k\} \end{array} \right\}}{\displaystyle\sum_{r=1}^{N} T\left\{ \begin{array}{l} \overset{N}{\underset{\substack{k=1 \\ k:\overline{z}^k_j \neq \overline{z}^r_j}}{T}} N\{\tau^k\}, \\ \overset{N}{\underset{\substack{k=1 \\ k:\overline{z}^k_j=\overline{z}^r_j}}{T}} S\{N\{\tau^k\}, \tau^k\} \end{array} \right\}}, \tag{33}$$

The final decision can be derived as follows:

$$\begin{cases} x \in \omega_j & \text{if } \overline{z}_j > z_{\text{IN}} \\ x \notin \omega_j & \text{if } \overline{z}_j < z_{\text{OUT}} \\ \text{undefined} & \text{if } z_{\text{OUT}} \leq \overline{z}_j \leq z_{\text{IN}}, \end{cases} \tag{34}$$

where the numbers z_{IN} and z_{OUT} are fixed thresholds such that $1 > z_{\text{IN}} \geq z_{\text{OUT}} > 0$.

5. Description of a logical RNFC

In this section we study the neuro-fuzzy classifier proposed in Section 4 in a specific situation i.e., when incomplete information about the object is available. Let us assume what follows:

- A classifier is set up and developed for n features of classified objects. Q denotes the set of all features of objects used in the course of system developing.

- In the course of classification of object x, only the values of $n_D \leq n$ features are known. $D \subseteq Q$ denotes the set of features whose values are known. $G = Q \setminus D$ denotes the set of features whose values are unknown.

The classifier defined in Section 4 does not work in such a situation. Our goal is to define a special version of a neuro-fuzzy classifier which could work in the described situation. In the proposed classifier we use a rough-fuzzy set, so the system is called a rough-neuro-fuzzy classifier.

It is obvious that, if we assume various values of unknown features $\mathbf{v}_G$, we obtain various values of $\overline{z}_j$ as the output of the neuro-fuzzy classifier. In most cases it is not possible to test all values of vector $\mathbf{v}_G$ and check if all the obtained values of $\overline{z}_j$ satisfy one of the conditions given in (34). However, it is enough to find the smallest possible value of $\overline{z}_j$ denoted as $\underline{\overline{z}_j}$ and the highest one denoted as $\overline{\overline{z}_j}$. This notation refers to that of rough sets and rough-fuzzy sets. The value $\underline{\overline{z}_j}$ is the membership degree of object x to the $\widetilde{D}$-lower approximation of set ω_j:

$$\mu_{\underline{\widetilde{D}}\omega_j}(x) = \underline{\overline{z}_j}, \tag{35}$$

and $\overline{\overline{z}}_j$ is the membership degree of object x to the $\widetilde{D}$-upper approximation of set ω_j:

$$\mu_{\overline{\overline{D}}\omega_j}(x) = \overline{\overline{z}}_j. \tag{36}$$

Theorem 1. (Rough membership of a class) *Let us consider the neuro-fuzzy classifiers defined by Eqns. (31) and (33). We assume that the classifier has been developed for n input features and all parameters of it, i.e., fuzzy sets A_j^r and B_j^r like in the rules (22), are fixed. Moreover, at the time of classification of object x only the values of $n_D \le n$ features are known, i.e., features from set $D \subseteq Q$. Then the lower and upper approximation of the membership of object x to class ω_j is given by*

• *S-implications, D-implications and R-implication:*

$$\underline{z}_j = \frac{\displaystyle\sum_{\substack{r=1 \\ r:\overline{z}_j^r=1}}^{N} \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k=0}}^{N} N\{\tau_{\mathrm{L}}^k\}}{\displaystyle\sum_{r=1}^{N} \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k \ne \overline{z}_j^r}}^{N} N\{\tau_{\mathrm{L}}^k\}}, \tag{37}$$

$$\overline{z}_j = \frac{\displaystyle\sum_{\substack{r=1 \\ r:\overline{z}_j^r=1}}^{N} \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k=0}}^{N} N\{\tau_{\mathrm{U}}^k\}}{\displaystyle\sum_{r=1}^{N} \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k \ne \overline{z}_j^r}}^{N} N\{\tau_{\mathrm{U}}^k\}}, \tag{38}$$

• *QL-implications:*

$$\underline{z}_j = \frac{\displaystyle\sum_{\substack{r=1 \\ r:\overline{z}_j^r=1}}^{N} T\left\{\begin{array}{c} \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k=0}}^{N} N\{\tau_{\mathrm{L}}^k\}, \\ \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k=1}}^{N} \min_{t=\mathrm{L,U}} S\{N\{\tau_t^k\},\tau_t^k\} \end{array}\right\}}{\displaystyle\sum_{r=1}^{N} T\left\{\begin{array}{c} \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k \ne \overline{z}_j^r}}^{N} N\{\tau_{\mathrm{L}}^k\}, \\ \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k=\overline{z}_j^r \\ k:\overline{z}_j^k=0}}^{N} \max_{t=\mathrm{L,U}} S\{N\{\tau_t^k\},\tau_t^k\}, \\ \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k=\overline{z}_j^r \\ k:\overline{z}_j^k=1}}^{N} \min_{t=\mathrm{L,U}} S\{N\{\tau_t^k\},\tau_t^k\} \end{array}\right\}}, \tag{39}$$

$$\overline{z}_j = \frac{\displaystyle\sum_{\substack{r=1 \\ r:\overline{z}_j^r=1}}^{N} T\left\{\begin{array}{c} \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k=0}}^{N} N\{\tau_{\mathrm{U}}^k\}, \\ \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k=1}}^{N} \max_{t=\mathrm{L,U}} S\{N\{\tau_t^k\},\tau_t^k\} \end{array}\right\}}{\displaystyle\sum_{r=1}^{N} T\left\{\begin{array}{c} \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k \ne \overline{z}_j^r}}^{N} N\{\tau_{\mathrm{U}}^k\}, \\ \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k=\overline{z}_j^r \\ k:\overline{z}_j^k=0}}^{N} \min_{t=\mathrm{L,U}} S\{N\{\tau_t^k\},\tau_t^k\}, \\ \mathop{T}_{\substack{k=1 \\ k:\overline{z}_j^k=\overline{z}_j^r \\ k:\overline{z}_j^k=1}}^{N} \max_{t=\mathrm{L,U}} S\{N\{\tau_t^k\},\tau_t^k\} \end{array}\right\}}, \tag{40}$$

where A_{L}^r and A_{U}^r are defined as follows:

$$\tau_{\mathrm{L}}^r = \begin{cases} \mu_{\underline{\widetilde{D}}A^r} & \text{if } \overline{z}_j^r=1, \\ \mu_{\overline{\overline{D}}A^r} & \text{if } \overline{z}_j^r=0, \end{cases} \tag{41}$$

and

$$\tau_{\mathrm{U}}^r = \begin{cases} \mu_{\overline{\overline{D}}A^r} & \text{if } \overline{z}_j^r=1, \\ \mu_{\underline{\widetilde{D}}A^r} & \text{if } \overline{z}_j^r=0. \end{cases} \tag{42}$$

Proof. Equations (37) and (38) are a direct consequence of Eqn. (31), so the proof will point out the correctness of Eqns. (41) and (42). The symbol τ_{L}^r as well as τ_{U}^r can be replaced by the lower approximation of fuzzy set A^r, i.e., $\underline{\widetilde{D}}A^r$ or upper approximation $\overline{\widetilde{D}}A^r$, where

$$\underline{\widetilde{D}}A^r \le \overline{\widetilde{D}}A^r. \tag{43}$$

The correctness of the choice described by Eqns. (41) and (42) arises due to

$$\left.\frac{\partial \overline{z}_j}{\partial \tau_{\mathrm{L}}^l}\right|_{l:\ \overline{z}_j^l=0} \le 0 \tag{44}$$

and

$$\left.\frac{\partial \overline{z}_j}{\partial \tau_{\mathrm{L}}^l}\right|_{l:\ \overline{z}_j^l=1} \ge 0, \tag{45}$$

as well as

$$\left.\frac{\partial \overline{\overline{z}}_j}{\partial \tau_{\mathrm{U}}^l}\right|_{l:\ \overline{z}_j^l=0} \le 0 \tag{46}$$

and

$$\left.\frac{\partial \overline{\overline{z}}_j}{\partial \tau_{\mathrm{U}}^l}\right|_{l:\ \overline{z}_j^l=1} \ge 0. \tag{47}$$

The same procedure can be used for classifiers based on QL-implications defined by (39) and (40). ∎

It is easy to see that, $\underline{z}_j < \overline{z}_j$.

The final decision can be derived as follows:

$$
\begin{cases}
x \in \omega_j & \text{if } \underline{z}_j \geq z_{\text{IN}} \text{ and } \overline{z}_j > z_{\text{IN}}, \\
x \notin \omega_j & \text{if } \underline{z}_j < z_{\text{OUT}} \text{ and } \overline{z}_j \leq z_{\text{OUT}}, \\
\text{rather } x \in \omega_j & \text{if } z_{\text{IN}} > \underline{z}_j \geq z_{\text{OUT}} \text{ and } \overline{z}_j > z_{\text{IN}}, \\
\text{rather } x \notin \omega_j & \text{if } \underline{z}_j < z_{\text{OUT}} \text{ and } z_{\text{OUT}} < \overline{z}_j \leq z_{\text{IN}}, \\
\text{undefined} & \text{otherwise,}
\end{cases}
$$

(48)

where $\underline{z}_j = \mu_{\underline{\tilde{D}\omega_j}}(x)$ is the lower approximation of the membership degree of object x to class ω_j and $\overline{z}_j = \mu_{\overline{\tilde{D}\omega_j}}(x)$ is its upper approximation. The two numbers (thresholds) z_{IN} and z_{OUT} are fixed such that $1 > z_{\text{IN}} \geq z_{\text{OUT}} > 0$.

Example 1. Let us assume that we have some neuro-fuzzy classifier and its parameters are $n = 4$, $N = 4$, $m = 1$. Moreover, we assume that for some sample with all four input features known we obtain the following membership degree of fuzzy sets used in the antecedent parts of rules:

$$\mu_{A_1^1}(v_1) = 0.9,$$
$$\mu_{A_2^1}(v_2) = 0.9,$$
$$\mu_{A_3^1}(v_3) = 0.5,$$
$$\mu_{A_4^1}(v_1) = 0.9,$$
$$\mu_{A_1^2}(v_2) = 0.9,$$
$$\mu_{A_2^2}(v_2) = 0.1,$$
$$\mu_{A_3^2}(v_3) = 0.5,$$
$$\mu_{A_4^2}(v_4) = 0.9,$$
$$\mu_{A_1^3}(v_1) = 0.1,$$
$$\mu_{A_2^3}(v_2) = 0.9,$$
$$\mu_{A_3^3}(v_3) = 0.5,$$
$$\mu_{A_4^3}(v_4) = 0.1,$$
$$\mu_{A_1^4}(v_1) = 0.1,$$
$$\mu_{A_2^4}(v_2) = 0.1,$$
$$\mu_{A_3^4}(v_3) = 0.5,$$
$$\mu_{A_4^4}(v_4) = 0.1.$$

Then the activation levels of individual rules are $\tau^1 = 0.3645$, $\tau^2 = 0.0405$, $\tau^3 = 0.0045$ and $\tau^4 = 0.0005$. When the centres of the fuzzy sets used in the consequence parts of rules are $\overline{z}^1 = 1$, $\overline{z}^2 = 0$, $\overline{z}^3 = 0$, $\overline{z}^4 = 1$ and we use any S-implication, then we obtain the output value $\overline{z} = 0.5571$. When we use $z_{\text{IN}} = z_{\text{OUT}} = 0.5$, we can interpret the output value as a conclusion $x \in \omega$.

Now, let us assume that the value of the second feature is unknown. We cannot use values $\mu_{A_2^1}(v_2)$, $\mu_{A_2^2}(v_2)$,

$\mu_{A_2^3}(v_2)$ and $\mu_{A_2^4}(v_2)$. Instead, we can use the lower and upper values of an appropriate membership function, i.e.,

$$\mu_{\underline{\tilde{D}}A_2^1}(v_2) = 0.2,$$
$$\mu_{\overline{\tilde{D}}A_2^1}(v_2) = 0.8,$$
$$\mu_{\underline{\tilde{D}}A_2^2}(v_2) = 0.2,$$
$$\mu_{\overline{\tilde{D}}A_2^2}(v_2) = 0.8,$$
$$\mu_{\underline{\tilde{D}}A_2^3}(v_2) = 0.2,$$
$$\mu_{\overline{\tilde{D}}A_2^3}(v_2) = 0.8,$$
$$\mu_{\underline{\tilde{D}}A_2^4}(v_2) = 0.2,$$
$$\mu_{\overline{\tilde{D}}A_2^4}(v_2) = 0.8.$$

Therefore, we obtain

$$\tau_{\text{L}}^1 = \mu_{\underline{\tilde{D}}A^1}(\mathbf{v}) = 0.081,$$
$$\tau_{\text{U}}^1 = \mu_{\overline{\tilde{D}}A^1}(\mathbf{v}) = 0.324,$$
$$\tau_{\text{L}}^2 = \mu_{\overline{\tilde{D}}A^2}(\mathbf{v}) = 0.324,$$
$$\tau_{\text{U}}^2 = \mu_{\underline{\tilde{D}}A^2}(\mathbf{v}) = 0.081,$$
$$\tau_{\text{L}}^3 = \mu_{\overline{\tilde{D}}A^3}(\mathbf{v}) = 0.004,$$
$$\tau_{\text{U}}^3 = \mu_{\underline{\tilde{D}}A^3}(\mathbf{v}) = 0.001,$$
$$\tau_{\text{L}}^4 = \mu_{\underline{\tilde{D}}A^4}(\mathbf{v}) = 0.001,$$
$$\tau_{\text{U}}^4 = \mu_{\overline{\tilde{D}}A^4}(\mathbf{v}) = 0.004.$$

Finally, we have a pair of output values, i.e., $\underline{z} = 0.4569$ and $\overline{z} = 0.5431$. We can interpret it as the conclusion which says that it is undefined if the object belongs to class ω or not.

When the values of feature v_3 are unavailable then we obtain other values of the output, e.g., $\underline{z} = 0.5111$ and $\overline{z} = 0.6275$. We can interpret this as the conclusion which says that the object belongs to class ω. ♦

Tables 1–4 show the results of rough neuro-fuzzy classifier performance for a few samples and their interpretation.

6. Experimental results

The three benchmarks from the UCI repository (Mertz and Murphy, n.d.) were used to test the performance of the proposed classifiers. They are

- Glass Identification (GI),

- Pima Indians Diabetes (PID),

- Breast Cancer Wisconsin (BCW).

In the case of all three datasets, the same experimental procedure was performed. The first step is preparing the dataset for the 5-fold cross validation procedure. The datasets were randomly divided into five subsets. In individual cross validation stages, the four subsets will be used as a training sequence and one—as a testing sequence. The training sequences were used to generate the fuzzy sets and rules, i.e., to develop the classifier. The LEM-2 algorithm is used (Grzymala-Busse, 1989; 1992).

Two types of classifiers were tested. The first one is based on S-implications and D-implications, and de-

Table 1. Result of classification for selected samples of the PID database (eight known features).

Known attributes		
$v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8$		
$\underline{z}_{\text{cancer}}$	$\overline{z}_{\text{cancer}}$	conclusion
0.28	0.28	$x \notin \omega_{\text{cancer}}$
0.13	0.13	$x \notin \omega_{\text{cancer}}$
0.29	0.29	$x \notin \omega_{\text{cancer}}$
0.35	0.35	$x \notin \omega_{\text{cancer}}$

Table 2. Result of classification for selected samples of the PID database (six known features).

Known attributes		
v_1,v_2,v_3,v_6,v_7,v_8		
$\underline{z}_{\text{cancer}}$	$\overline{z}_{\text{cancer}}$	conclusion
0.17	0.65	undefined
0.02	0.59	undefined
0.23	0.73	undefined
0.26	0.66	undefined

Table 3. Result of classification for selected samples of the PID database (four known features).

Known attributes		
v_2,v_4,v_5,v_8		
$\underline{z}_{\text{cancer}}$	$\overline{z}_{\text{cancer}}$	conclusion
0.01	0.73	undefined
0.00	0.49	$x \notin \omega_{\text{cancer}}$
0.01	0.48	$x \notin \omega_{\text{cancer}}$
0.00	0.49	$x \notin \omega_{\text{cancer}}$

Table 4. Result of classification for selected samples of the PID database (two known features).

Known attributes		
v_4,v_8		
$\underline{z}_{\text{cancer}}$	$\overline{z}_{\text{cancer}}$	conclusion
0.01	0.99	undefined
0.00	0.80	undefined
0.01	0.48	$x \notin \omega_{\text{cancer}}$
0.00	0.88	undefined

fined by Eqns. (37) and (38). The second one is defined by Eqns. (39) and (40) i.e., based on QL-implications. In the presented experiment only one QL-implication was used, based on the probabilistic t-conorm

$$S(a,b) = a + b - ab \qquad (49)$$

and any t-norm. In all tested classifiers we used the simplest negation, i.e.,

$$N(a) = 1 - a. \qquad (50)$$

The final decisions were derived as defined in (48) for $z_{\text{IN}} = z_{\text{OUT}} = 0.5$.

Both types of classifiers were tested using separate training and testing sequences. Each test is repeated for all possible sets of input features as known (the number of known features). Let us suppose that in a database four features are defined four. In such a case, 15, i.e., $2^4 - 1$, experiments are performed, as in Table 5.

In the glass classification problem (Mertz and Murphy, n.d.) we have 214 samples described by nine features. They are: v_1—refractive index (RI) and the participation of eight components (v_2—Na_2O, v_3—MgO, v_4—Al_2O_3, v_5—SiO_2, v_6—K_2O, v_7—CaO, v_8—BaO, v_9—Fe_2O_3). The samples of glass belong to two classes: the window glass (ω_1) and other kind of glass (ω_2). As we can read in (Mertz and Murphy, n.d.), the study of the classification of glass types was motivated by criminological investigations. At the scene of a crime, the glass left can be used as an evidence if it is correctly classified. The average results are presented in Tables 6–9.

In PID, each of the 768 records of the database describes a representative of the Pima Indians heritage who

Table 5. List of experiments using a hypothetical dataset with four input attributes.

Experiment number	No. of known features	List of known features
1	4	v_1, v_2, v_3, v_4
2		v_1, v_2, v_3
3	3	v_1, v_2, v_4
4		v_1, v_3, v_4
5		v_2, v_3, v_4
6		v_1, v_2
7		v_1, v_3
8	2	v_1, v_4
9		v_2, v_3
10		v_2, v_4
11		v_3, v_4
12		v_1
13	1	v_2
14		v_3
15		v_4

is healthy or ill (two classes) and is characterized by eight features (v_1—number of times pregnant, v_2—plasma glucose concentration in an oral glucose tolerance test, v_3—diastolic blood pressure (mm Hg), v_4—triceps skin fold thickness (mm), v_5—2-hour serum insulin (mu U/ml), v_6—body mass index (weight in kg/(height in m)^2), v_7—diabetes pedigree function, v_8—age in years) (Mertz and Murphy, n.d.). The averaged results are presented in Tables 10–13.

In the BCW data set we have 683 fully described records about patients. Each record is described by nine attributes, i.e., clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, mitoses. The average results are presented in Tables 14–17.

Table 6. S-implication classifier performance for training the GI sequence.

No. of known features	Classifications [%]		
	correct	miss	incorrect
9	95.2	0.0	4.8
8	76.4	22.4	1.2
7	49.2	50.7	0.2
6	22.0	78.0	0.0
5	7.2	92.8	0.0
4	1.9	98.1	0.0
3	0.5	99.5	0.0
2	0.1	99.9	0.0
<2	0.0	100.0	0.0

Table 7. S-implication classifier performance for testing the GI sequence.

No. of known features	Classifications [%]		
	correct	miss	incorrect
9	88.4	0.0	11.6
8	72.8	22.7	4.6
7	47.5	51.0	1.5
6	21.4	78.2	0.3
5	7.0	92.9	0.1
4	1.8	98.2	0.0
3	0.4	99.6	0.0
2	0.1	99.9	0.0
<2	0.0	100.0	0.0

Table 8. QL-implication classifier performance for training the GI sequence.

No. of known features	Classifications [%]		
	correct	miss	incorrect
9	93.0	0.0	7.0
8	48.4	51.3	0.3
7	17.5	82.5	0.0
6	6.2	93.8	0.0
5	2.3	97.7	0.0
4	0.8	99.2	0.0
3	0.3	99.7	0.0
2	0.1	99.9	0.0
<2	0.0	100.0	0.0

Table 9. QL-implication classifier performance for testing the GI sequence.

No. of known features	Classifications [%]		
	correct	miss	incorrect
9	87.0	0.0	13.0
8	47.6	50.0	2.4
7	17.4	82.3	0.3
6	6.0	93.9	0.0
5	2.1	97.8	0.0
4	0.7	99.3	0.0
3	0.2	99.8	0.0
2	0.1	99.9	0.0
<2	0.0	100.0	0.0

Table 10. S-implication classifier performance for training the PID sequence.

No. of known features	Classifications [%]		
	correct	miss	incorrect
8	87.0	0.0	13.0
7	37.2	61.5	1.1
6	6.4	93.6	0.0
5	0.2	99.8	0.0
<5	0.0	100.0	0.0

Table 11. S-implication classifier performance for testing the PID sequence.

No. of known features	Classifications [%]		
	correct	miss	incorrect
8	70.2	0.0	29.8
7	30.6	63.7	5.8
6	5.8	93.9	0.4
5	0.2	99.8	0.0
<5	0.0	100.0	0.0

Table 12. QL-implication classifier performance for training the PID sequence.

No. of known features	Classifications [%]		
	correct	miss	incorrect
8	86.0	0.0	14.0
7	7.1	92.9	0.0
6	0.5	99.5	0.0
<6	0.0	100.0	0.0

As we have seen in all presented results, the proposed classifier gives the answer only when it is sure enough. When the input knowledge is limited, then the classifiers refrains from giving any answer. It protects us from making mistakes.

To compare the proposed method with others, Tables 18–23 show the result of classification using the k-nn algorithm. As we can see, when the input features are missing, then the number of mistakes grows.

7. Conclusions

In this paper the modification of a neuro-fuzzy classifier has been proposed. Based on the rough fuzzy sets the scope of employing the classifier has been extended to the cases when some input information is missing. The

Table 13. QL-implication classifier performance for testing the PID sequence.

No. of known features	Classifications [%]		
	correct	miss	incorrect
8	66.8	0.0	33.2
7	5.5	93.7	0.9
6	0.4	99.6	0.0
<6	0.0	100.0	0.0

Table 14. S-implication classifier performance for training the BCW sequence.

No. of known features	Classifications [%]		
	correct	miss	incorrect
9	97.4	0.0	2.6
8	80.3	19.0	0.8
7	43.7	56.1	0.2
6	14.9	85.1	0.0
5	2.9	97.1	0.0
4	0.3	99.7	0.0
<4	0.0	100.0	0.0

Table 15. S-implication classifier performance for testing the BCW sequence.

No. of known features	Classifications [%]		
	correct	miss	incorrect
9	94.8	0.0	5.2
8	76.8	20.9	2.2
7	41.5	57.5	1.0
6	14.0	85.6	0.4
5	2.6	97.3	0.1
4	0.3	99.7	0.0
<4	0.0	100.0	0.0

Table 16. QL-implication classifier performance for training the BCW sequence.

No. of known features	Classifications [%]		
	correct	miss	incorrect
9	96.8	0.0	3.2
8	33.6	66.4	0.1
7	7.8	92.2	0.0
6	2.0	98.0	0.0
5	0.4	99.6	0.0
<5	0.0	100.0	0.0

Table 17. QL-implication classifier performance for testing the BCW sequence.

No. of known features	Classifications [%]		
	correct	miss	incorrect
9	91.6	0.0	8.4
8	30.9	68.1	1.0
7	6.9	92.7	0.3
6	1.5	98.3	0.2
5	0.3	99.7	0.1
4	0.0	99.9	0.0
<4	0.0	100.0	0.0

Table 18. k-nn classifier performance for training the GI sequence.

No. of known features	Classifications [%]	
	correct	incorrect
9	90.8	9.2
8	91.2	8.8
7	91.4	8.6
6	91.5	8.5
5	91.3	8.7
4	90.8	9.2
3	89.8	10.2
2	87.9	12.1
1	76.0	24.0

Table 19. k-nn classifier performance for testing the GI sequence.

No. of known features	Classifications [%]	
	correct	incorrect
9	87.8	12.2
8	88.9	11.1
7	89.4	10.6
6	89.6	10.4
5	89.4	10.6
4	88.9	11.1
3	87.9	12.1
2	86.1	13.9
1	82.9	17.1

new solution can select these input instances when, despite the lack of data, classification is available. For other instances, the classifier does not give a questionable answer. Unfortunately, the experiments shown in Section 6 demonstrate that the number of unclassified instances increases dramatically when the input information is incomplete. However, the number of mistakes does no grow.

Competitive solutions, e.g., based on the k-nn algorithm, give usually much more correct classifications, but the number of mistakes also increases. The main benefit of the proposed modification is protection against mistakes in the case of missing input data. The complexity of the proposed solution with S-implications, D-implications and R-implications is similar to that of other fuzzy and neuro-fuzzy systems, which has been broadly discussed, e.g., (Marin *et al.*, 2008; Jin, 2000). In the case of QL-implications, the complexity is 2^N times higher.

The paper concludes a trilogy by the author (Nowicki, 2008; 2009). It presents rough-neuro-fuzzy classifiers that work with missing features but having various properties and structures due to various methods of reasoning and defuzzification used. Users can choose the solution appropriate for his or her problem.

Acknowledgment

Table 20. k-nn classifier performance for training the PID sequence.

No. of known features	Classifications [%]	
	correct	incorrect
8	79.0	21.0
7	77.7	22.3
6	76.9	23.1
5	76.1	23.9
4	75.1	24.9
3	74.0	26.0
2	72.2	27.8
1	68.4	31.7

Table 21. k-nn classifier performance for testing the PID sequence.

No. of known features	Classifications [%]	
	correct	incorrect
8	72.4	27.6
7	71.9	28.1
6	70.9	29.1
5	69.8	30.2
4	68.6	31.4
3	67.3	32.7
2	65.3	34.7
1	62.9	37.1

References

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford.

Broekhoven, E. V. and Beats, B. D. (2006). Fast and accurate center of gravity defuzzification of fuzzy system outputs defined on trapezoidal fuzzy partitions, *Fuzzy Sets and Systems* **157**(7): 904–918.

Chan, L. S., Gilman, J. A. and Dun, O. J. (1976). Alternative approaches to missing values in discriminant analysis, *Jour-*

Table 22. k-nn classifier performance for training the BCW sequence.

No. of known features	Classifications [%]	
	correct	incorrect
9	97.2	2.8
8	97.0	3.0
7	96.8	3.2
6	96.6	3.4
5	96.3	3.7
4	96.0	4.0
3	95.3	4.7
2	93.7	6.3
1	88.3	11.7

Table 23. k-nn classifier performance for testing the BCW sequence.

No. of known features	Classifications [%]	
	correct	incorrect
9	97.0	3.0
8	96.8	3.2
7	96.4	3.6
6	96.3	3.7
5	96.0	4.0
4	95.5	4.5
3	94.7	5.3
2	93.1	6.9
1	88.1	11.9

nal of the American Statistical Association **71**(356): 842–844.

Cooke, M., Green, P., Josifovski, L. and Vizinho, A. (2001). Robust automatic speech recognition with missing and unreliable acoustic data, *Speech Communication* **34**(3): 267–285.

Czogała, E. and Łęski, J. (2000). *Fuzzy and Neuro-Fuzzy Intelligent Systems*, Physica-Verlag, Heidelberg/New York, NY.

Dixon, J. K. (1979). Pattern recognition with partly missing data, *IEEE Transactional on Systems, Man and Cybernetics* **9**(10): 617–621.

Driankov, D., Hellendoorn, H. and Reinfrank, M. (1993). *An Introduction to Fuzzy Control*, Springer-Verlag, Berlin/Heidelberg.

Dubois, D. and Prade, H. (1990). Rough fuzzy sets and fuzzy rough sets, *International Journal of General Systems* **17**(2–3): 191–209.

Dubois, D. and Prade, H. (1992). Putting rough sets and fuzzy sets together, *in* R. Słowiński (Ed.), *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*, Kluwer, Dordrecht, pp. 203–232.

Duda, R. O., Hart, P. E. and Stork, D. G. (2001). *Pattern Classification*, A Wiley-Interscience Publication, John Wiley & Sons, Inc., New York, NY/Chichester/Weinheim/Brisbane/Singapore/Toronto.

Fodor, J. C. (1991). On fuzzy implication operators, *Fuzzy Sets and Systems* **42**(3): 293–300.

Fogel, D. B. (1995). *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence*, IEEE Press, New York, NY.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company Inc, Reading, MA.

Grzymala-Busse, J. W. (1989). An overview of the LERS1 learning systems, *Proceedings of the 2nd International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Tullahoma, TN, USA*, pp. 838–844.

Grzymala-Busse, J. W. (1992). LERS—A system for learning from examples based on rough sets, *in* R. Slowinski (Ed.), *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*, Kluwer, Dordrecht, pp. 3–18.

Jin, Y. (2000). Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement, *IEEE Transactions on Fuzzy Systems* **8**(2): 212–221.

Kecman, V. (2001). *Learning and Soft Computing, Support Vector Machines, Neural Networks and Fuzzy Logic Models*, The MIT Press, Cambridge, MA.

Klement, E. P., Mesiar, R. and Pap, E. (2000). *Triangular Norms*, Kluwer Academic Publishers, Dordrecht.

Kuncheva, L. I. (2000). *Fuzzy Classifier Design. Studies in Fuzziness and Soft Computing*, Physica-Verlag, Heidelberg.

Lee, K. M. and Kwang, D. H. (1994). A fuzzy neural network model for fuzzy inference and rule tuning, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **2**(3): 265–277.

Lin, C. T. and Lee, G. C. S. (1991). Neural-network-based fuzzy logic control and decision system, *IEEE Transactions on Computers* **40**(12): 1320–1336.

Little, R. J. A. and Rubin, D. B. (2002). *Statistical Analysis with Missing Data, 2nd Edn*, Wiley-Interscience, New York, NY.

Marin, N., Molina, C., Serrano, J. and Vila, M. (2008). A complexity guided algorithm for association rule extraction on fuzzy datacubes, *IEEE Transactions on Fuzzy Systems* **16**(3): 693–714.

Mas, M., Monserrat, M., Torrens, J., and Trillas, E. (2007). A survey on fuzzy implication functions, *IEEE Transactions on Fuzzy Systems* **15**(6): 1107–1121.

Mertz, C. J. and Murphy, P. M. (n.d.). UCI respository of machine learning databases, http://www.ics.uci.edu/pub/machine-learning-databases.

Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin/Heidelberg/New York, NY.

Morin, R. L. and Raeside, D. E. (1981). A reappraisal of distance-weighted k-nearest neighbor classification for pattern recognition with missing data, *IEEE Transactions on Systems, Man and Cybernetics* **11**(3): 241–243.

Nauck, D., Klawonn, F. and Kruse, R. (1997). *Foundations of Neuro-Fuzzy Systems*, Wiley, Chichester.

Nowicki, R. (2000). *The Neuro-Fuzzy Systems which Realizes Various Methods of Fuzzy Reasoning*, Ph.D. thesis, AGH University of Science and Technology, Cracow, (in Polish).

Nowicki, R. (2004). Rough sets in the neuro-fuzzy architectures based on non-monotonic fuzzy implications, *in* L. Rutkowski, J. Siekmann, R. Tadeusiewicz and L.A. Zadeh (Eds), *Artificial Inteligence and Soft Computing—ICAISC 2004*, Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin/Heidelberg, Vol. 3070, pp. 518–525.

Nowicki, R. (2008). On combining neuro-fuzzy architectures with the rough set theory to solve classification problems with incomplete data, *IEEE Transactions on Knowledge Data Engineering* **20**(9): 1239–1253.

Nowicki, R. (2009). Rough-neuro-fuzzy structures for classification with missing data, *IEEE Transactions on Systems, Man and Cybernetics B* **39**(6): 1334–1347.

Nowicki, R. and Rutkowska, D. (2000). Neuro-fuzzy architectures based on yager implication, *Proceedings of the 5th Conference on Neural Networks and Soft Computing, Zakopane, Poland*, pp. 353–360.

Patel, A. V. and Mohan, B. M. (2002). Some numerical aspects of center of area defuzzification method, *Fuzzy Sets and Systems* **132**(3): 401–409.

Pawlak, Z. (1982). Rough sets, *International Journal of Information and Computer Science* **11**(341): 341–356.

Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning About Data*, Kluwer, Dordrecht.

Pawlak, Z. (2002). Rough sets, decision algorithms and Bayes' theorem, *European Journal of Operational Research* **136**(1): 181–189.

Pedrycz, W. and Bargiela, A. (2002). Granular clustering: A granular signature of data, *IEEE Transactions on Systems, Man and Cybernetics B* **32**(2): 212–224.

Polkowski, L. (2002). *Rough Sets. Mathematical Foundation*, Physica-Verlag, Heidelberg/New York, NY.

Renz, C., Rajapakse, J. C., Razvi, K. and Liang, S. K. C. (2002). Ovarian cancer classification with missing data, *Proceedings of the 9th International Conference on Neural Information Processing, ICONIP'02, Singapore*, Vol. 2, pp. 809–813.

Rutkowska, D. and Nowicki, R. (2000a). Implication-based neuro-fuzzy architectures, *International Journal of Applied Mathematics and Computer Science* **10**(4): 675–701.

Rutkowska, D. and Nowicki, R. (2000b). New neuro-fuzzy architectures, *Proceedings of the International Conference on Artificial and Computational Intelligence for Decision, Control and Automation in Engineering and Industrial Applications, AcIDcA'2000, Monastir, Tunisia*, pp. 82–87.

Rutkowska, D., Nowicki, R. and Rutkowski, L. (2000). Neuro-fuzzy architectures with various implication operators, *The State of the Art in Computational Intelligence—Proceedings of the International Symposium on Computational Intelligence (ISCI 2000), Kosice, Slovakia*, pp. 214–219.

Rutkowski, L. and Cpałka, K. (2003). Flexible neuro-fuzzy systems, *IEEE Transactions on Neural Networks* **14**(3): 554–574.

Rutkowski, L. and Cpałka, K. (2005). Designing and learning of adjustable quasi-triangular norms with applications to neuro-fuzzy systems, *IEEE Transactions on Fuzzy Systems* **13**(1): 140–151.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery* **2**(2): 121–167.

Tanaka, M., Kotokawa, Y. and Tanino, T. (1996). Patern classification by stochastic neural network with missing data, *IEEE International Conference on System, Man and Cybernetics, Beijing, China*, Vol. 1, pp. 690–695.

Wang, L. X. (1994). *Adaptive Fuzzy Systems and Control*, PTR Prentice Hall, Englewood Cliffs, NJ.

Yager, R. R. and Filev, D. P. (1994). *Essentials of Fuzzy Modeling and Control*, John Wiley and Sons, New York, NY.

Yao, J. T. and Yao, Y. Y. (2002). Induction of classification rules by granular computing, *in* J.J. Alpigini, J.F. Peters, J.F. Skowron and N. Zhang (Eds), *Rough Sets and Current Trends in Computing,* Lecture Notes in Artificial Intelligence, Springer, Berlin/Heidelberg, Vol. 2475, pp. 331–338.

Zadeh, L. A. (1965). Fuzzy sets, *Information and Control* **8**(3): 338–353.

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning—Part 1, *Information Sciences* **8**(3): 199–249.

Żurada, J. M. (1992). *Introduction to Artificial Neural Systems*, West Publishing Company, St. Paul, MN.

Robert Nowicki was born in Częstochowa, Poland, in 1972. He received the M.Sc. degree in electrical engineering from the Technical University of Częstochowa in 1997 and the Ph.D. degree in computer engineering from the AGH University of Science and Technology in 2000. Since 2000 he has been an assistant professor at the Department of Computer Engineering, Technical University of Częstochowa. His research interests include fuzzy systems, artificial neural networks, genetic algorithms, rough sets, and hybrid methods. He has published over 40 technical papers in journals, conference proceedings and book chapters.