World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:7, 2009

Categorical Missing Data Imputation Using Fuzzy Neural Networks with Numerical and Categorical Inputs

Pilar Rey-del-Castillo, and Jesús Cardeñosa

Open Science Index, Computer and Information Engineering Vol:3, No:7, 2009 publications.waset.org/7285/pdf

***Abstract*—**There are many situations where input feature vectors are incomplete and methods to tackle the problem have been studied for a long time. A commonly used procedure is to replace each missing value with an imputation. This paper presents a method to perform categorical missing data imputation from numerical and categorical variables. The imputations are based on Simpson's fuzzy min-max neural networks where the input variables for learning and classification are just numerical. The proposed method extends the input to categorical variables by introducing new fuzzy sets, a new operation and a new architecture. The procedure is tested and compared with others using opinion poll data.

***Keywords*—**Classifier, imputation techniques, fuzzy systems, fuzzy min-max neural networks.

I. Introduction

MISSING information in data sets is a more than common scenario. There are many grounds for missing information in real-world applications: automatic equipment sensor errors or failures, optional data fields in medical files, refusals by respondents to answer questions compromising their privacy in surveys, etc. Since most statistical and data mining techniques assume that data is complete and there is no missing information, methods to tackle the problem [1], [2] have been under development for a long time. There are a number of references, basically about non-response in surveys [3], [4].

A commonly used procedure is to replace each missing variable value with an estimated value or imputation obtained from the non-missing values of other variables in the same unit. There are different methods to perform these imputations depending on the type of variable with missing data and on the type of auxiliary variables. Here the imputation of categorical variables from other numerical and categorical variables is studied.

In this article, performing imputations based on a neuro-fuzzy classifier, a supervised learning method that takes a hybrid neural networks and fuzzy systems approach, is proposed.

This is one of the most popular hybridizations in the artificial intelligence literature because it combines the merits of the neural and fuzzy approaches. It has the innate benefits of neural networks –like massive parallelism and robustness– and, at the same time, uses fuzzy logic to model vague or qualitative knowledge and convey uncertainty [5].

The proposed classifier is based on Simpson's fuzzy min-max neural networks [6], [7] where the input variables for learning and classification are numerical only. The presented method extends the input to categorical variables by introducing new fuzzy sets, a new operation and a new architecture, allowing for greater flexibility and wider application.

The new procedure is applied to non-response imputation in an opinion poll. The microdata (the set of the respondents' individual answers to the questions) of this type of poll are especially suitable for evaluating the method, since they include numerical and categorical variables.

The article is organized as follows. Section II describes the context, the data used and the polls it is taken from, emphasizing the non-response problem to be solved. Section III presents the architecture and operation of fuzzy min-max neural networks as a starting point for the new classifier, whereas section IV shows the new method based on new fuzzy sets from which new networks –and their architecture and operation– are defined. Section V presents the results of the experiment in the described context. They improve upon the results of applying traditional methods on the same data set.

II. Problem Context: Political Opinion Polls

A frequent procedure used to collect information about a population is to make a survey. When the questions refer to individual opinions or attitudes, these surveys are known as opinion polls [8]. These polls have proven to be an especially fast and easy-to-use tool, because they simplify the most technical phases of the survey process. As in most surveys, there is usually total or partial non-response –when the respondent fails to answer one or more of the questions, respectively–. The procedure for total non-response is usually addressed at the sampling design stage, and this paper focuses on partial non-response.

Partial non-response is generally solved by imputing values to the missing variables from the answers of other respondents and from the non-missing variables of the same individual.

P. Rey-del-Castillo is with the Centro de Investigaciones Sociológicas, 28014 Madrid Spain (phone: +3491-580-7689; fax: +3491-580-7619; e-mail: prey@ cis.es).

J. Cardeñosa is with the Artificial Intelligence Department, Universidad Politécnica de Madrid, Spain (e-mail: carde@fi.upm.es).

Open Science Index, Computer and Information Engineering Vol:3, No:7, 2009 publications.waset.org/7285/pdf

However, speed is more important in polls than accuracy, and the usual way of dealing with non-response is to add the "Don't know/Not applicable" category and treat it like any other category. This is not a highly recommendable method because it causes problems at the results analysis stage [3], but it is widely applied in polls due to its straightforwardness.

In election polls, though, there is one variable –*which political party do you intend to vote for in the elections* (voting intention, from now on)– for which the above method is not good enough, and missing values have to be imputed. Elsewhere we presented a paper where fuzzy control procedures were used to estimate voting intention in an electoral poll [9]. It stressed the potential of using methods to automatically obtain fuzzy set membership functions. This is what we propose to do now using neural networks. In this work a new fuzzy min-max neural networks classifier is applied to impute missing voting intentions from the answers to other questions in the same survey. To evaluate its operation, we selected poll number 2750 from the Sociological Research Center's (institution responsible for making opinion polls for the Spanish Public Administration) catalog. The survey refers to the general elections held in Spain in 2008, containing 13.358 interviews with an answer to the voting intention question. The chosen poll contains questions with different types of variables:

- *Quantitative variables.* Questions answered by entering a numerical value. They include questions referring to *ideological self-location* (the result of asking respondents to place themselves ideologically on a scale of 1 to 10, 1 being the extreme left and 10 the extreme right). Other possibilities are the *rating on a scale of 0 to 10 of three specific political figures*, the *likelihood to vote,* and the *likelihood to vote for three specific political parties*.
- *Ordered categorical variables.* Questions answered by entering categories that are so well ordered that they are easy and straightforward to transform into quantitative variables. They refer to government and opposition party ratings. The answer categories are "*very good*", "*good*", "*fair*", "*bad*" and "*very bad*", which we transform into the values *1, 0.75, 0.5, 0.25 and 0*, respectively, assuming they are ordered equidistantly. As we will see, they should take values within the unit interval like the membership functions of fuzzy sets.
- *Categorical variables with non-ordered categories*. Questions including voting intention and similar, such as *vote memory* (party the respondent voted for at the last general election), the *Autonomous Community*, *which of the likely candidates the respondent would prefer to see as president of the government*, *how sure/definite the respondent's voting intention is*, *the political party the respondent tips to win* and *the political party the respondent would prefer to win*.

Although missing values are found in all the above variables, this paper focuses on the imputation of the categorical *voting intention* variable only. The method most used nowadays to impute votes in opinion polls is to make predictions from logistic regressions with other variables. Besides, different procedures based on neural networks have been used to impute numerical variables from other likewise numerical values [10]-[12]. There is no knowledge of their use for imputing categorical variables from other numerical and categorical variables, as proposed in this paper.

The new procedure presented here is an extension of Gabrys and Bargiela's model [13], [14] –which is, in turn, based on an earlier model by Simpson [6], [7]–. The new procedure is an improvement on its predecessors since, unlike the earlier models –which exclusively admit numerical input data–, it accepts numerical and categorical data as inputs. Also it significantly improves upon earlier results, as shown in the experiment outlined in section V. To give a better understanding of the proposed method set out in section IV, the fuzzy min-max neural networks is described in section III.

III. Gabrys and Bargiela's Model

The original fuzzy min-max neural networks algorithm was introduced in two articles by Simpson [6], [7]. It is a classification method that partitions the joint input variables space using nonlinear boundaries. Here, a later version that includes some improvements by Gabrys and Bargiela [13], [14] is outlined.

A. Classification Model

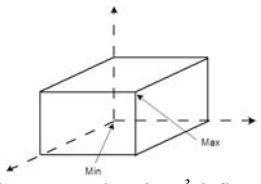


Fig. 1 Hyperbox in R^3 defined from its min and max points

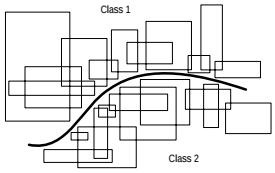


Fig. 2 Fuzzy min-max hyperboxes along the boundary

The n input variables must be numerical and the output variable –the variable to be imputed– is to be categorical. The operation is based on the hyperbox fuzzy sets defined in the n-dimensional pattern space. A hyperbox in R^n is a Cartesian product of closed intervals on the real line and is completely defined by its minimum and maximum points, as shown in the three-dimensional example in Fig. 1. Although it is possible to use hyperboxes with an arbitrary range of values in any dimension, min-max networks only use values that range from 0 to 1.

Thus, the input space is the n-dimensional unit cube $I^n = [0,1]\times[0,1]\times\ldots\times[0,1]$. The hyperbox fuzzy set B_j is defined by the ordered set:

$$B_j = \left\{x, v_j, w_j, b_j\left(x, v_j, w_j\right)\right\}, \forall x \in I^n \quad (1)$$

where $v_j = \left(v_{j1},\ldots,v_{jn}\right)$ is the hyperbox minimum, $w_j = \left(w_{j1},\ldots,w_{jn}\right)$ is the maximum, and $b_j\left(x, v_j, w_j\right)$ is the membership function, where all patterns within the hyperbox have full class membership. Each class or classification

Open Science Index, Computer and Information Engineering Vol:3, No:7, 2009 publications.waset.org/7285/pdf

category matches one of the different values of the variable to be imputed, and it is the union of hyperbox fuzzy sets $C_k = \bigcup_{j \in K} B_j$, where K is the set of indexes of the kth class hyperbox fuzzy sets. Fig. 2 shows an example of how the hyperboxes are aggregated to form nonlinear boundaries in a two-class I^2 classification problem. The first stage for classifying an input pattern is to calculate its membership function of each class as the maximum of its membership functions of each one of the hyperboxes defining this class (the maximum is the selected fuzzy union operator). The next stage is to classify –in our case, impute– the point as the category corresponding to the class with the greatest membership function.

One of Gabrys and Bargiela's improvements was to allow hyperboxes and not just numerical points as input patterns. In this case, each input is specified by a vector x_h, $h=1, 2, ..., M$, where $x_h = \left[x_h^l \; x_h^u\right]$ is the hth input hyperbox defined by its minimum vector $x_h^l = \left(x_{h1}^l, x_{h2}^l, \ldots, x_{hn}^l\right)$ and its maximum vector $x_h^u = \left(x_{h1}^u, x_{h2}^u, \ldots, x_{hn}^u\right)$. When x_h^l and x_h^u are equal, the hyperbox shrinks to a point. The membership function of the hyperbox fuzzy set B_j for an input x_h is defined as

$$b_j(x_h) = \min_{i=1,\ldots,n} \left\{ \min \left[\left(1 - g\left(x_{hi}^u - w_{ji}, \gamma\right)\right), \left(1 - g\left(v_{ji} - x_{hi}^l, \gamma\right)\right) \right] \right\} \quad (2)$$

where γ is a parameter regulating how fast the membership function decreases and g is the ramp-threshold function of two parameters:

$$g(x, \gamma) = \begin{cases} 1 & si \quad x.\gamma > 1 \\ x.\gamma & si \quad 0 \le x.\gamma \le 1 \\ 0 & si \quad x.\gamma < 0 \end{cases} \quad (3)$$

The function takes the value 1 –full membership– within the hyperbox and decays to zero as x_h moves away from the hyperbox. A two-dimensional example is shown in Fig. 3 for the hyperbox fuzzy set defined by the minimum $v_j = (0.4, 0.2)$, the maximum $w_j = (0.6, 0.4)$ and the parameter $\gamma = 3$.

The hyperboxes are incrementally trained by appropriately adjusting their number and volumes in a neural networks framework. This accounts for the name of fuzzy min-max neural networks. The network architecture and learning are described next.

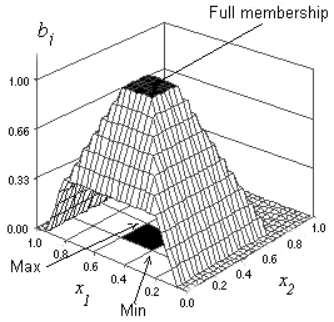


Fig. 3 Membership function of the hyperbox in I^2 defined by the minimum $v_j = (0.4, 0.2)$, the maximum $w_j = (0.6, 0.4)$ and the parameter $\gamma = 3$

B. Network Architecture

The three-layer neural network implementing Gabrys and Bargiela's fuzzy min-max neural classifier is shown in Fig. 4. Its topology is modified to meet the problem requirements. The input layer has $2n$ nodes, two for each of the n input vector dimensions corresponding to the minimums $\left(x_{hi}^l\right)$ and the maximums $\left(x_{hi}^u\right)$ of the input hyperboxes. Each intermediate layer node represents a hyperbox fuzzy set, where the connections with the input layer are the hyperbox fuzzy set minimum $\left(v_{ji}\right)$ and maximum $\left(w_{ji}\right)$ points, and the activation function is the above hyperbox membership function (2).

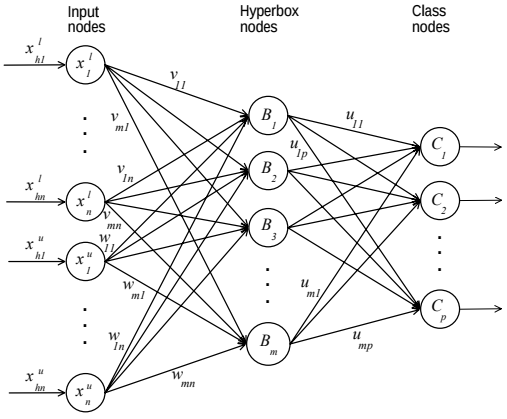


Fig. 4 Three-layer neural network implementing the fuzzy min-max neural network classifier

Fig. 5 shows the jth node of the intermediate layer in more detail. The connections between the second- and third-layer nodes are binary values, whose expression is

$$u_{jk} = \begin{cases} 1 & if \; B_j \; is \; a \; hyperbox \; for \; class \; C_k \\ 0 & otherwise \end{cases} \quad (4)$$

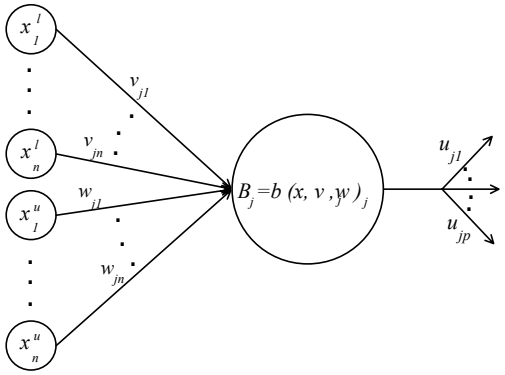


Fig. 5 Implementation of the *j*th node of the intermediate layer. It has *2n* connections with the first layer, corresponding to the mins and maxs in each dimension of the *j*th hyperbox fuzzy set, and *p* with the last layer, one for each of its class nodes

where B_j is the *j*th intermediate layer node and C_k is the *k*th output layer node. The result of this last node represents the membership degree of input x_h to class *k*. The activation function for each output layer node *p* is the fuzzy union of the hyperbox membership functions according to the expression $c_k = \max_{j=1}^{m} b_j u_{jk}$. The classifier result for x_h is the class *k* with the greatest c_k value. The values for the connections are adjusted using the learning algorithm described next.

C. Learning Algorithm

The learning set consists of *M* ordered pairs $\{x_h, d_h\}$, *h=1, ..., M*, where $x_h = \left[x_h^l \; x_h^u\right]$ is the *h*th input defined by its min $x_h^l = \left(x_{h1}^l, x_{h2}^l, \ldots, x_{hn}^l\right)$ and max $x_h^u = \left(x_{h1}^u, x_{h2}^u, \ldots, x_{hn}^u\right)$ points, and $d_h \in \{1, 2, \ldots, p\}$ is the index of one of the *p* classes. The process begins with the input of an ordered pair searching the hyperbox with the greatest membership degree, belonging to the same class and including or allowing expansion to include x_h. If none of the hyperboxes satisfies the conditions, then a new hyperbox B_k for the input is created, adjusted and labeled by making $class(B_k)=d_h$. This learning process forms classes that are non-linearly separable. This way existing classes can be refined over time, and new classes can be added without retraining. The hyperbox expansion can lead to an overlap between them. It is constrained by a user-defined parameter θ, $(0 \le \theta \le 1)$ where $|w_{ji} - v_{ji}| \le \theta, \forall \; i=1, \ldots n$. The overlap is not a problem when it occurs between hyperboxes representing the same class. When there is an overlap between hyperboxes that represent different classes, it is solved using a contraction process following the principle of minimal adjustment. The contraction process only eliminates the overlap between those portions of the hyperbox fuzzy sets from separate classes that have full membership, allowing overlap between non-unit-valued portions of each of the hyperbox fuzzy sets. The bounds between two classes are the points with an equal membership function for both classes. In summary, the fuzzy min-max learning algorithm is a four-step process:

1. Search for the closest expandable hyperbox (if necessary)
2. Expand hyperbox
3. Test for hyperbox overlap
4. Contract hyperbox

It is repeated for each training input point.

IV. Model with Input of Categorical Variables

In contrast to the fuzzy min-max neural networks classifier, the proposed procedure considers categorical as well as numerical variables as input. The problem with a categorical variable input is that there is no measure of distance between the different values or categories from which hyperbox fuzzy sets membership functions can be defined. We will describe the new model according to the same framework as Gabrys and Bargiela's model. The basic process will be:

- Define distances between categories
- Define hyperbox fuzzy sets in categorical variables
- Extend network architecture and operation

A. Defining Distances between Categories

TABLE I

		Religion				
		Cath.	Prot	Muslim	Others	Total
Region	North	186	27	27	60	300
	West	48	6	18	48	120
	Center	63	109	148	32	352
	East	59	12	8	22	101
	South	31	63	94	21	209
Total		387	217	295	183	1082

To define a distance between the categories of a categorical variable, the relation of this variable to the variable to be imputed, which we have also assumed to be categorical, is considered. To illustrate this idea, Table I shows an example with the contingency table for the categorical variables *region* and *religion*. Table II is calculated from Table I by just dividing the value of each cell by its row total. The vector $(q_1, \ldots, q_p)$ in each row of Table II contains the response rates for the *religion* categories in this *region*, referred to as the region's *religious* profile.

TABLE II

		Religion			
		Cath.	Prot.	Muslim	Others
Region	North	0.62	0.09	0.09	0.2
	West	0.4	0.05	0.15	0.4
	Center	0.18	0.31	0.42	0.09
	East	0.58	0.12	0.08	0.22
	South	0.15	0.3	0.45	0.1
Total		0.36	0.2	0.27	0.17

Open Science Index, Computer and Information Engineering Vol:3, No:7, 2009 publications.waset.org/7285/pdf

Open Science Index, Computer and Information Engineering Vol:3, No:7, 2009 publications.waset.org/7285/pdf

To define distances between *regions*, their profiles are looked at, i.e. the *center* and *south* regions have similar profiles, (0.18, 0.31, 0.42, 0.09) and (0.15, 0.30, 0.45, 0.10), respectively. This means that religion is similarly distributed in these regions. The profiles for the *north* and *east* regions are also similar, albeit different from the *center* and *south* regions, whereas the *west* region is very different to the others. It could be said that, regarding religion, the *center* and *south* regions are closer to each other than to all the others; the *north* and *east* are also close, and so on. The category profiles are points of the p-dimensional space R^p belonging to the hyperplane defined by $q_1 + \ldots + q_p = 1$. The distances between the profiles in this space can be used to define the distances between the categories. In this paper we considered two distances:

$$\text{Euclidean distance: } d_1(a_i, a_j) = \sqrt{\sum_{k=1}^{p} (p_{ik} - p_{jk})^2} \quad (5)$$

$$\text{Logarithmic distance: } d_2(a_i, a_j) = \sum_{k=1}^{p} \left| \log p_{ik} - \log p_{jk} \right| \quad (6)$$

where a_i, a_j are the categories and (p_{ik}), (p_{jk}), $k=1, \ldots, p$, are the corresponding profiles. As the proportions forming the profiles take values between 0 and 1, the logarithmic distance is considered in an attempt to prevent proportionally short distances between high values from overdominating the calculations. Besides, to standardize and use the distances in the context of fuzzy set membership functions, we divide them by their maximum:

$$c_k(a_i, a_j) = \frac{d_k(a_i, a_j)}{\max_{i,j} d_k(a_i, a_j)}, , k = 1, 2 \quad (7)$$

Correspondence analysis [15], for example, exploits this idea of distance between profiles. We discuss next how the distances can be used for inputting categorical variables in fuzzy min-max neural networks.

B. Defining Hyperbox Fuzzy Sets in Categorical Variables

Once the distances between the categories are defined, the next step is to define the hyperbox fuzzy sets in the categorical dimensions. As the minimum and maximum points determine the hyperbox in the numerical dimensions, it is assumed that, in the ith categorical dimension, it is determined by two categories e_{ji} and f_{ji} (which can also be equal) with a full membership function (equal to 1). In any other category a_{ki}, this ith dimension membership function takes the value

$$b_{ji}(a_{h\,i}) = \min\left(1 - c(a_{hi}, e_{ji}), 1 - c(a_{hi}, f_{ji})\right) \quad (8)$$

where function c refers to any of the normalized distances previously defined in (7), and the size of the hyperbox in each dimension is limited by a user-defined parameter η, ($0 \leq \eta \leq 1$), where $c(e_{ji}, f_{ji}) \leq \eta$. Fig. 6 is an example of the symmetric distance function $c(a_k, a_l)$ between the five categories of a variable and the membership function $b_j(a_k)$ obtained for the jth hyperbox that is determined by the two full membership categories $e_j = a_3$ and $f_j = a_5$.

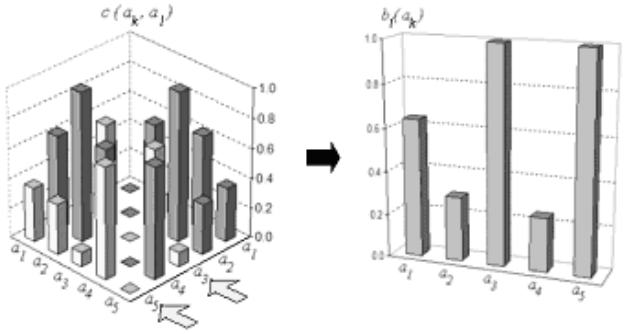


Fig. 6 The symmetric distance function between categories $c(a_k, a_l)$ and the membership function $b_j(a_k)$ of the hyperbox defined by categories $e_j = a_3$ and $f_j = a_5$

When there are numerical and categorical variables, **we define the B_j hyperbox membership function –of all the dimensions–** by

$$b_j(x_h, a_h) = \min\left\{ \min_{i=1,\ldots,n} \left[\min\left(1 - g(x_{hi}^u - w_{ji}, \gamma), 1 - g(v_{ji} - x_{hi}^l, \gamma)\right)\right], \min_{i=n+1,\ldots,n+r} \left[\min\left(1 - c_i(a_{hi}, e_{ji}), 1 - c_i(a_{hi}, f_{ji})\right)\right]\right\} \quad (9)$$

where n is the number of numerical variables and r is the number of categorical variables; g is the ramp-threshold function defined in (3) and c_i, $i = n+1, \ldots, n+r$, are the normalized distances defined in (7) for the categorical dimensions; $x_h = \left[x_h^l \; x_h^u\right]$ is the numerical input defined by its vectors of minimum $\left(x_{hi}^l\right)$ and maximum $\left(x_{hi}^u\right)$ points; $a_h = (a_{h\,n+1}, \ldots, a_{h\,n+r})$ is the categorical input vector; $v_{j\,i}$ is the minimum and $w_{j\,i}$ is the maximum of the jth hyperbox in the ith numerical dimension, $i = 1, \ldots, n$; and $e_{j\,i}$, $f_{j\,i}$ are the two categories defining hyperbox B_j in the ith categorical dimension $i = n+1, \ldots, n+r$.

C. Extend network architecture and operation

The above membership function treats the categorical variables in the same manner as it processes the numerical variables, where the inputs are categories in the first case and numerical hyperboxes in the second: the distances c_i play the role of functions g and they are combined by the same fuzzy operators. This naturally extends the neural network operation. Fig. 7 shows the new network architecture including both types of variables, and Fig. 8 is the detail of an intermediate layer node. The only difference from Gabrys and Bargiela's network is the input layer, where, apart from the $2n$

Open Science Index, Computer and Information Engineering Vol:3, No:7, 2009 publications.waset.org/7285/pdf

numerical variable nodes, there are *r* additional nodes for the input categories, each having two connections with the second-layer nodes –one for each category $e_{j\,i}$, $f_{j\,i}$ defining the B_j hyperbox– .

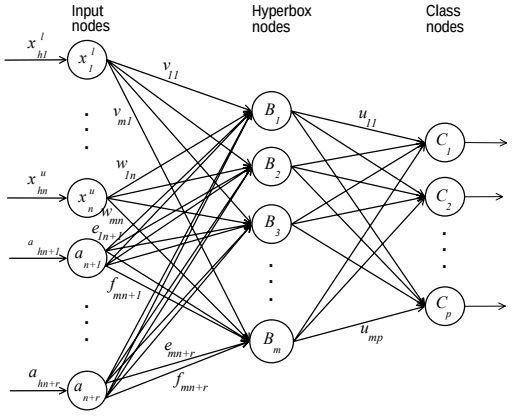


Fig. 7 Topology of the fuzzy min-max neural network implementing the new classifier, which admits numerical and categorical inputs

The second layer maintains a node for each hyperbox, which has numerical and categorical dimensions in this case. A network input takes the form $\left(x^l_{h1},\ldots,\ x^l_{hn},x^u_{h1},\ldots,x^u_{hn},a_{hn+1},\right.$ $\left.\ldots,\ a_{hn+r}\right)$, where x^l_{hi} are the minimums and x^u_{hi} are the maximums of the input hyperboxes in dimension *i*, *(i = 1,..., n)*, and $a_{h\,i}$ are the input categories in dimension *i*, *(i = n+1,..., n+r)*. The second-layer activation function is the membership function defined in (9), and the connections between the second-layer and first-layer categorical nodes are the two categories e_{ji} and f_{ji}, *(*e_{ji} *,* f_{ji} *=* a_1*, ...,* a_p*)* defining the B_j hyperbox in dimension *i*, *(i = n+1, ..., n+r)*. Its numerical node connections are the same B_j hyperbox minimums v_{ji} and maximums w_{ji}. Finally, like the original network, the third layer has a node for each one of the variable classification categories, and its connections with the intermediate layer are the same $u_{j\,k}$ as defined in (4).

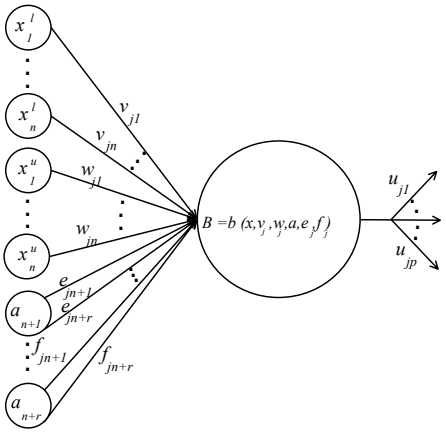


Fig. 8 Detail of the nodes connected with the *j*th node of the intermediate layer

Learning aims to establish the connections v_{ji}, w_{ji}, e_{ji} and f_{ji}, that is, the hyperboxes defining each class. The first step – taken only once– is to calculate the distances between the categories of categorical variables and the resulting membership function, as described above. This is followed by the iterative process to determine and update the connection values. This process is repeated for each input and has the same phases as the original network, with some minor changes:

1. Search for the expandable hyperbox with the greatest membership function
2. Expand hyperbox
3. Test hyperboxes for overlap
4. Contract hyperboxes according to the test result

The expansion step must establish the two classes defining the hyperboxes for each categorical dimension, plus the hyperbox minimums and maximums for the numerical dimensions. For this purpose, the input a_h is taken when there is no preset category in either of these dimensions. If there is a preset category, it is checked that the distance between the two does not exceed the size limits $c\ (e_{ji}, f_{ji}) \le \eta$, $(0 \le \eta \le 1)$ before it is taken as the second category.

When the overlap test result is positive, that is, when there is a non-empty overlap between portions with full membership representing different classes, the hyperboxes are contracted into a single dimension following the minimum change principle, beginning with the categorical dimensions. In one of these dimensions, the overlapping category of the recently expanded or created hyperbox is treated to change for another one reducing the hyperbox size, that is, another category closer to the first category defining the hyperbox. If this is possible, it is replaced –eliminating the overlap– and, if not, contracting in another dimension is tried, moving on to the numerical dimensions when there are no more categorical dimensions left. In this case, the contraction is performed as defined for the original network [9], distributing the overlapping space between the two hyperboxes.

Finally, the new network operates similarly to its predecessor in terms of classification: imputation is made by assigning the category corresponding to the class with the greatest membership function.

V. Case Study: Application to Voting Intention Imputation in a Political Poll

In this section, the new networks are applied to voting intention variable imputation in opinion poll number 2750 archived at the Sociological Research Center. Their performance is then compared with other classical methods. An evaluation criterion that is frequently used in the supervised classification procedures area is used for comparisons: the correctly imputed rate, that is, the percentage of imputed values that exactly match the original data over the 13358 inputs with non-missing voting intention. A 10-fold cross-validation is performed, partitioning the test data into ten parts (folds). A single fold is retained as the validation

Open Science Index, Computer and Information Engineering Vol:3, No:7, 2009 publications.waset.org/7285/pdf

data for testing the model, where the remaining nine folds are used as training data. The cross-validation process is then repeated 10 times with each of the 10 folds, and the results are averaged to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once. This procedure provides non-biased estimations of the correctly imputed rate.

Eleven categories are taken for the voting intention variable, including the most important political parties, *"Blank vote"*, *"Abstention"* and a category of *"Others"*. This would appear to be quite a good granularity level for obtaining reliable proportions for nationwide voting intention, whereas a larger granularity would make the problem tougher. The above sixteen numerical and ordered and non-ordered categorical variables are used as classifier inputs.

As already mentioned, the procedure most used nowadays for voting intention imputations in opinion polls is to make predictions from logistic regressions with other variables, and it will be taken as a baseline for comparison. The 10-fold cross-validation results of the data set with this type of regression and the sixteen variables (generated using SAS/STATS software, Version 9.1.3 of the SAS System for Windows. Copyright © 2002-2003 by SAS Institute Inc., Cary, NC, USA.) give a correctly imputed rate of 63.05%. Note that the likelihood equation for a logistic regression model does not always have a finite solution, making it difficult to estimate model parameters. Sometimes there is a non-unique maximum on the boundary of the parameter space at infinity. The existence, finiteness, and uniqueness of maximum likelihood estimates for the logistic regression model depend on the patterns of data points in the observation space [16]. When there is a complete or quasi-complete separation, there exist infinite estimations, and only if there is an overlap of sample points do unique maximum likelihood estimates exist. In this case, there is the possibility of separation because of the great many variables and categories and the output models are questionable. Another problem with the use of logistic regression is that units with missing values in the input variables are deleted, reducing the learning set size.

To make an additional comparison, using the same fuzzy min-max neural network classifier, another distance frequently used with categorical variables is considered: if a_i, a_j are two categories, then $c_3\ (a_h\ ,\ a_j\ ) = 1 - \delta_{h\ j}$, where $\delta_{h\ j}$ is the Kronecker delta. In this case, the hyperbox membership function is defined by

$$b_j(x_h, a_h) = \min\left\{ \min_{i=1,\ldots,n} \left[\min\left(1 - f(x^u_{hi} - w_{ji}, \gamma_i),\ \left(1 - f(v_{ji} - x^l_{hi}, \gamma_i)\right)\right], \min_{i=n+1,\ldots,n+r} \left[1 - c_3(a_{hi}, e_{ji})\right] \right\} \quad (10)$$

where $e_{j\ i}$ is the only category defining the hyperbox B_j in that dimension i.

The experiment has been performed implementing a classifier for each one of the three membership functions corresponding to the three distances. As the designed networks have some user-defined parameters for adjustment (the maximum numerical hyperbox size θ, the numerical membership function decreasing parameter γ and the maximum categorical hyperbox size η), estimations have been made for the set of parameter combinations resulting from γ = 0.5, 1.5, 2.5, 3.5, 4.5, θ = 0.15, 0.25, 0.35, 0.45, 0.55 and η = 0.15, 0.25, 0.35, 0.45, 0.55 (η makes no sense with the Kronecker distance). Tables III, IV, and V show the correctly imputed rates with the 10-fold cross-validation for the five parameter combinations returning the best results for each membership function.

TABLE III
EUCLIDEAN DISTANCE

γ	θ	η	% correctly imputed
2.5	0.55	0.55	86.06
2.5	0.55	0.35	85.93
2.5	0.55	0.45	85.93
1.5	0.45	0.55	85.91
2.5	0.55	0.25	85.91

TABLE IV
LOGARITHMIC DISTANCE

γ	θ	η	% correctly imputed
0.5	0.35	0.55	85.06
0.5	0.45	0.15	85.04
0.5	0.25	0.55	84.9
0.5	0.55	0.15	84.88
0.5	0.35	0.45	84.86

TABLE V
KRONECKER DISTANCE

γ	θ	% correctly imputed
0.5	0.55	72.86
0.5	0.45	72.65
0.5	0.35	72.46
0.5	0.25	72.42
1.5	0.55	72.02

As the learning set order may have an impact on the results, the validation process was repeated several times with various randomizations of the input set. They resulted in similar rates, thereby confirming the method's robustness.

The tables merit a number of remarks:

1. The correctly imputed rates for the Euclidean and the logarithmic distance are significantly greater than for the Kronecker distance and logistic regression. They are up around 13 percentage points over the first, and 22 percentage points over the second. The results range –up to 86%, even with a great many classification categories–

is much better than what is usually achieved in similar polls.

2. There is no clear difference between the behavior of the Euclidean and logarithmic distances, and the logarithmic distance does not appear to solve potential problems stemming from proportionally short distances between high values. The question requires more thorough investigation before a distance is selected.
3. Gabrys and Bargiela propose the use of different parameters θ and γ for each numerical dimension. The same parameters were used here, and we were able to improve results by varying the θ, γ *and* η thresholds in each dimension.
4. The procedure presented here is specially suited for the case of a relatively high number of categories for imputation, as opposed to the more commonly dealt with case of binary variables with just two categories.
5. Note that the proposed neuro-fuzzy classifier is suitable when the number of input variables –numerical and categorical– is high. In the case of missing values in input data sets, logistic regression estimations take into account only the complete data inputs. If there are a lot of variables all with non-response, the number of inputs may decrease dangerously. The proposed procedure always uses all the available data in the most efficient way, and the more variables there are, the better the results will be. Using the method, the select variables step can be eliminated, and this leads to more automatic imputation.

This paper presents very early results, and the next step will be to test the method on other public repository files.

REFERENCES

[1] J. L. Schafer, *Analysis of Incomplete Data*, Chapman & Hall, London, 1997.
[2] P. Allison, *Missing Data*, Sage Publications, Inc, 2002.
[3] R. J. Little, and D. B. Rubin, *Statistical Analysis with Missing Data,* 2nd ed. , John Wiley and Sons, New York, 2002.
[4] A. P. Dempster, and D. B. Rubin, "Incomplete data in sample surveys" in W. G. Madow, I. Olkin, and D. B. Rubin, Eds., *Sample Surveys, Vol. II: Theory and Annotated Bibliography*, New York, Academic Press, 1983.
[5] S. Mitra, S. K. Pal, and P. Mitra, "Data mining in soft computing framework: a survey", *IEEE Transactions on Neural Networks*, vol. 13, issue 1, pp. 3-14, Jan. 2002.
[6] P. K. Simpson, "Fuzzy min-max neural networks- Part 1: classification", *IEEE Transactions on Neural Networks*, vol. 3, Sep. 1992, pp. 776-786.
[7] P. K. Simpson, "Fuzzy min-max neural networks- Part 2: clustering", *IEEE Transactions on Fuzzy Systems*, vol. 1, pp. 32-45, Feb. 1993.
[8] D. R. Cox, *Principles of Statistical Inference,* Cambridge University Press, 2006.
[9] J. Cardeñosa, and P. Rey-del-Castillo, "A fuzzy control approach for vote estimation", *Proceedings of the Fifth International Conference on Information Technologies and Applications*, vol. 1. Varna, Bulgaria, June 2007.
[10] M. Abdella, and T. Marwala, "The Use of Genetic Algorithms and Neural Networks to Approximate Missing Data in Database", *ICCC 2005, IEEE 3rd International Conference on Computational Cybernetics*, pp. 207-212, 2005.
[11] F. V. Nelwamondo, S. Mohamed, and T. Marwala, "Missing Data: A Comparison of Neural Network and Expectation Maximization Techniques", *Current Science*, vol. 93, no. 11, pp. 1514-1521, Dec. 2007.
[12] P. Lingras, M. Zhong, and S. Sharma, "Evolutionary Regression and Neural Imputations of Missing Values", *Soft Computing Applications in Industry, Studies in Fuzziness and Soft Computing Series*, vol. 226, Springer, Berlin/Heidelberg, pp. 151-163, 2008.
[13] B. Gabrys, and A. Bargiela, "General Fuzzy Min-Max Neural Network for Clustering and Classification", *IEEE Transactions on Neural Networks*, vol. 11, pp. 769-783, May 2000.
[14] B. Gabrys, "Neuro-Fuzzy Approach to Processing Inputs with Missing Values in Pattern Recognition Problems". *International Journal of Approximate Reasoning*, vol. 30, pp. 149-179, September 2002.
[15] M. J. Greenacre, *Theory and Applications of Correspondence Analysis*, Academic Press, London, 1984
[16] T. J. Santner, and D. E. Duffy, "A Note on A. Albert and J. A. Anderson's Conditions for the Existence of Maximum Likelihood Estimates in Logistic Regression Models", *Biometrika*, vol. 73, pp. 755-758, 1986.

Open Science Index, Computer and Information Engineering Vol:3, No:7, 2009 publications.waset.org/7285/pdf