1998 International Computer Symposium
Workshop on Artificial Intelligence
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

HOW TO HANDLE UNKNOWN INPUTS

Mercedes Fernandez Redondo,, and Carlos Hernández Espinosa

Department of Computers. Jaume-I University. Campus
Riu Sec. Edificio TI. 12071 Castellon. SPAIN.
Email:espinosa@inf.uji.es

ABSTRACT

This paper presents an empirical comparison among five different methods of handling missing inputs. Fifteen different problems are used for the comparison, and the performance of the methods is obtained for several percentages 0%, 5%, 10%, 20%, 30% and 40% of missing inputs in the problem. There is one method, which has the best performance in all the problems. The method is based on a generalization of Backpropagation to interval arithmetic, a training including missing inputs in the training set, and a codification during training of every missing input by an interval which lower and upper limits include the range of variation of the input.

1. INTRODUCTION

The problem of missing or unknown inputs can be briefly described as follows. During the training phase of a neural network, the information is inside a training set. Usually, every attribute of the training set is present, however, after the training phase, one or several attributes in an input instance may be missing. We have an incomplete vector but we still expect to get some information with the known part of our vector, we want to get the most likely class in such conditions. So the problem is how to codify the missing inputs, how to use them or how "to tell" the neural network that an input is missing.

This problem is quite typical in application areas like medical diagnosis, it is rather easy to design a neural network diagnosis system and to find that in a particular case one or several needed symptoms or tests are missing.

Several methods have been proposed in order to deal with this problem in the case of Multilayer Feedforward neural network architecture. We will review them briefly.

Ahmand and Tresp [1] discussed Bayesian techniques for extracting class probabilities for incomplete data with missing inputs, and presented [2] an approximation for dealing with missing inputs during training and recall based on the approximation of the input data distribution using Parzen windows and estimating the class probabilities given the known part of the incomplete vector.

Ishibuchi and Miyazaki, [3] and [4], proposed to train the neural network with missing data and codify the missing inputs by an interval.

Finally, in [5] it is introduced a method for deriving substitute values for the missing ones. This method is only valid in the case of binary inputs.

The objective of this paper is to present an empirical comparison among these methods and to show which one is the most efficient.

2. DIFFERENT METHODS

In this section we concisely describe the different methods for handling missing inputs which can be found in the bibliography.

2.1 Method 1

This method is a normal training including missing data as part of the training set. The missing inputs are substituted by the value 0.5 in the case of binary inputs, and in the case of real valued inputs, the known values are moved to the interval [0.25,1] and the missing values are codified by zero. This method is described in the reference [6].

2.2 Method 2

In [3] and [4] Ishibuchi et alt. proposed a new network architecture and training algorithm. The architecture is a generalization of Multilayer Feedforward to interval arithmetic, and the training algorithm is a generalization of Backpropagation for the case of interval input vectors. Both algorithms together allow to use interval vectors as part of the training set.

The method is based on a training using missing inputs as part of the training set and every missing input is codified by an interval, whose lower and upper limits include the full range of variation of the input. For instance, if the range of variation of the input is between 0 and 1, the interval is [0,1].

2.3 Method 3

This method is similar to method 2. It is also based on interval arithmetic, it needs to train including missing inputs in the training set and the codification of missing inputs is the same.

The difference is that the training algorithm is also a generalization of Backpropagation to interval arithmetic, but a different generalization. It was proposed by one of the authors of this paper in [7] and [8].

2.4 Method 4

Ahmad and Tresp [2], developed a method to estimate the classes probabilities given a value for the known inputs. After the estimation, we assign the input to the class with highest probability. There is no need to train including missing inputs.

The estimation is made according to the following equations:

$$P(class_i \mid x^c) = \frac{\sum_{k=1}^{N} NN_i(x^c, x^{u,k}) \cdot G(x^c; x^{c,k}; \sigma)}{\sum_{k=1}^{N} G(x^c; x^{c,k}; \sigma)} \quad (1)$$

$$G(x; x^k; \sigma) = \frac{1}{2 \cdot \pi \cdot \sigma} \cdot \exp\left(\frac{1}{2 \cdot \pi \cdot \sigma^2} \cdot \left\| x - x^k \right\|\right) \quad (2)$$

where $x^{c,k}$ are the components of the training data corresponding to the known inputs, and $NN(x^c, x^{u,k})$ is the network prediction which is obtained if we substitute the corresponding components of the training data for the unknown inputs.

2.5 Method 5

This method [5] is valid only in the case of binary inputs. Therefore, its applications are rather limited.

We use a normal training without missing inputs, and after training we substitute the missing inputs for a value which is calculated in the following way:

$$input_i = \frac{\sum_j r_{i,j} \cdot input_j \cdot m_j}{\sum_j r_{i,j} \cdot m_j} \quad (3)$$

where m_j is a vector of missing values indicators (0 when the corresponding entry in the vector is missing and 1 when it is present), j varies from 1 to the number of inputs, and r is calculated with the following equation:

$$r_{i,j} = \frac{\sum_{k=1}^{n} \overline{input_{k,j} \; XOR \; input_{k,j}}}{n} \quad (4)$$

3. EXPERIMENTAL RESULTS

We have applied the five methods to fifteen different real world problems, two of them with binary inputs.

The problems are public and can be found in the UCI repository of machine learning databases. In the following lines we include a brief description of the problems:

- *Credit Approval (CA):* This problem concerns credit card applications. It has 15 nominal and continuos attributes, 2 classes, 453 training instances and 200 test instances.
- *Pima Indians Diabetes (PI):* This problem has 8 attributes, 2 classes, 518 training instances and 250 test instances.
- *Image Segmentation (IS):* This problem has 19 continuo attributes, 7 classes, 1500 training instances and 811 test instances.
- *The Monk's Problems (MO1 and MO2):* We have used two of the three monk's problems. These problems were the basis of the first international comparison of learning algorithms. They are three problems with six attributes and two classes, 332 training instances and 100 test instances.
- *Display 1 (D1):* This problem contains seven attributes, the seven segments of a light-emitting LED display, and 10 classes, the set of decimal digits. Each attribute value has a 10% probability of having its value inverted.
- *Display 2 (D2):* It is the D1 problem, but additional seven irrelevant attributes are added to the input space. It has 900 training instances and 2000 test instances.
- *Cylinder Bands (CB):* The problem has 39 nominal and numeric attributes, two classes, 190 training instances and 87 testing instances.
- *Balance Scale (BS):* The attributes are the left weight and distance and the right ones, four numeric attributes. It has three classes tip to left, tip to the right and balanced. We have used 500 training samples and 125 testing samples.
- *Liver Disorders (LD):* Six numeric attributes, five of them are blood tests, two classes. It has 245 training samples and 100 testing samples.
- *Glass Identification (GI):* Classification among several types of glasses using the composition. It has ten numeric attributes, six classes, 150 samples for training and 64 samples for testing.
- *Heart Disease (HD):* Diagnosis of coronary artery disease. It has 13 attributes, two classes, 210 training data and 87 testing samples.
- *Voting Records (VR):* 1984 United States congressional voting records, the problem is to identify whether the person is democrat o republican using the answer to several questions. It has sixteen Boolean attributes, two classes, 335 training instances and 100 testing instances.
- *Wisconsin Diagnostic Breast Cancer (WD):* It has thirty inputs and two classes, 409 training samples and 160 testing samples.

For every problem, we randomly introduced a percentage of missing inputs, and we finally obtained six different sets of training data and six sets of test data for each problem, with a percentage of 0%, 5%, 10% 20%, 30% and 40% of missing inputs with respect to the total number of inputs. So, we have made experiments with a total of 90 different training sets.

After that, we trained six different networks using conjugate gradients for every problem and with every set of training data (percentages of 0%, 5%, 10%, 20%, 30% and

40%), and codifying the missing inputs according to the first method. In the test the performance measure was percentage correct and the classification rule was to choose the class with greater output.

Other six neural networks were trained by using the extension of Backpropagation to interval arithmetic and the codification of missing inputs described in method 2. In the test, we used the following classification rule, we utilise the upper limits of the interval output and we consider that the input is classified in the class whose upper limit is greater [3].

Finally, we trained other six networks with the extension of Backpropagation to interval arithmetic described in method 3. We measured the percentage correct with the same rule.

In every case, we worked out the average of the results of the six networks, so the results were calculated with an error and are statistically significant. They do not depend on the initial value of weights etc.

We will not presents the full results because of the lack of space, we present typical results and a few examples of the results.

In fig. 1 and 2 we have the results of method 1 for the problems PI and D1. As we can see from the figures, the results of this method presented a high variability, its performance strongly depends on the particular problem. The inclusion of unknown inputs in the training set makes the classification task more complex, and it seems that this classification task becomes too complex and Backpropagation can not solve the problem very well.

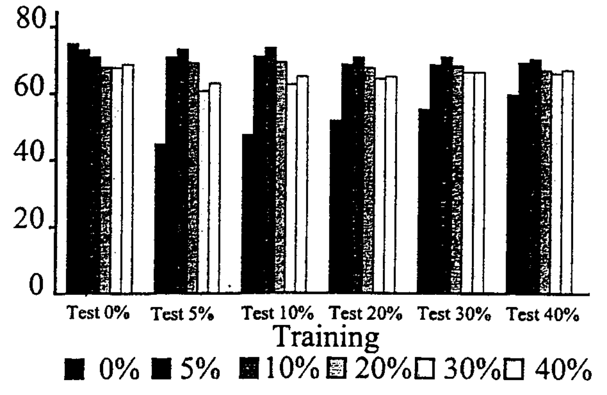

Fig. 1. Percentage correct for several test and training percentages of missing inputs, method 1, problem PI.

The performance decreases as the number of missing inputs in the training set increases, fig. 1. It also decreases as the number of missing inputs in the test increases.

The efficiency of this method, in general, was better than the one of method 4 and worse than the others.

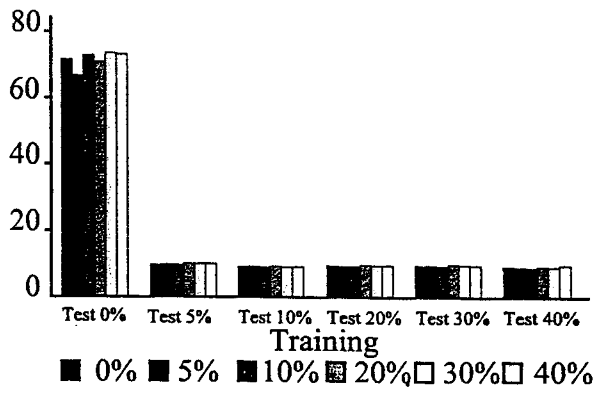

Fig. 2. Percentage correct for several test and training percentages of missing inputs, method 1, problem D1

In fig. 3 we have some results of method 2 for the problem CA. We can see that there is a maximum percentage of missing inputs in the training set, and over this percentage, which is 30% in fig. 3, the performance decreases suddenly and the method does not work. The maximum percentage depends on the problem.

Its performance is better than method 1, 4 and 5.

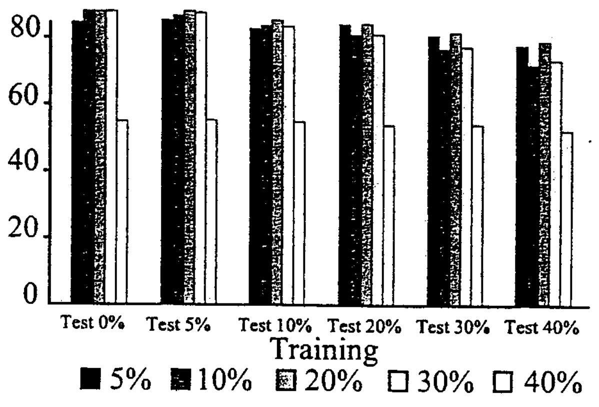

Fig. 3. Percentage correct for several test and training percentages of missing inputs, method 2, problem CA.

Method 3 has the same problem of maximum percentage of missing inputs in the training set. However, this method obtained the better results in the fifteen problems. We can see an example in fig. 4 and 5.

In figs. 4 and 5 we have some results for the method 4 and they can be compared with the other ones. This method got the worst results, there are several approximations in the equations of Ahmad and Tresp and they may be not very good.

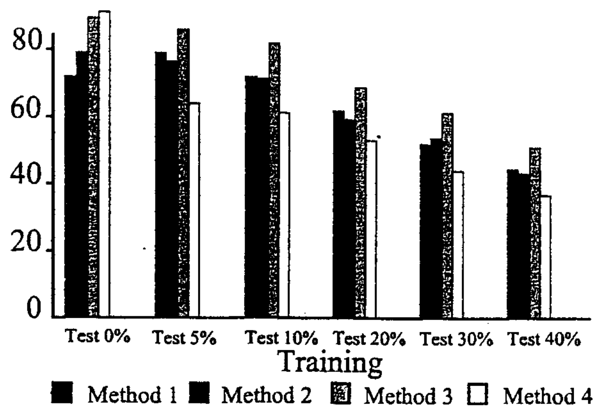

Fig. 4. Percentage correct for several tests of missing inputs and four methods, problem IS.

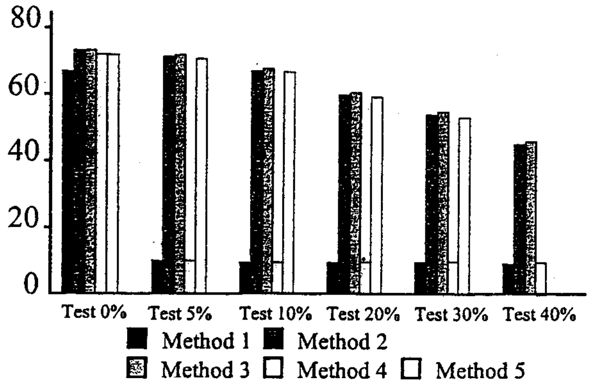

Fig. 5. Percentage correct for several tests of missing inputs and the five methods, problem D1.

In fig. 5 we can find some results of method 5. We can observe that its performance is similar to the one of method 3. However, it can be only applied to the case of binary inputs and therefore its applicability is rather limited.

4. CONCLUSIONS

We have presented an in depth empirical comparison of five different methods of handling missing inputs. The methods are tested in fifteen different classification problems with several values of missing inputs.

The problems are real world problems or realistic ones and public, they can be found in the UCI repository of machine learning databases.

The method number 3 of this paper got the better results, a missing input percentage of 5% in the training set can be appropriate in order to develop an application.

The results are statistically significant because they are the average of several neural networks.

5. REFERENCES

[1] Ahmad S., Tresp V.: Classification with missing and uncertain inputs. In: Proceedings of 1993 International Conference on Neural Networks. 1993, pp 1949-1954.

[2] Tresp V., Neuneier R., Ahmad S.: Efficient methods for dealing with missing data in supervised learning. In: Advances in Neural Information Processing Systems 7. 1995, pp. 689-696.

[3] Ishibuchi H., Miyazaki A., Known K., Tanaka H.: Learning from incomplete training data with missing values and medical application. In: Proceedings of 1993 International Conference on Neural Networks. 1993, pp 1871-1874.

[4] Ishibuchi H., Miyazaki A., Tanaka H.: Neural-network-based diagnosis systems for incomplete data with missing inputs. In: Proceedings of 1994 IEEE International Conference on Neural Networks. 1994, pp 3457-3460.

[5] Armitage W.D., Jien-Chung Lo: Enhancing the robustness of a Feedforward neural network in the presence of missing data. In: Proceedings of 1994 IEEE International Conference on Neural Networks. 1994, pp 836-839.

[6] Maren A., Harston C., Pap R.. Handbook of Neural Computing Applications. Academic Prees, 1990.

[7] Hernandez C.A., Espi J., Nakayama K.: A generalization of Backpropagation to Interval Arithmetic. In: Proceedings of the World Congress on Nerual Networks, (WCNN'93). 1993, vo. 4, pp 131-134.

[8] Hernandez C.A., Espi J., Nakayama K. Fernandez M.: Interval Arithmetic Backpropagation. In: Proceedings of the 1993 International Joint Conference on Neural Networks. 1993, vol. 1, pp 375-378.