# Intelligent Data Analysis Using Neuro-Fuzzy Approach

Kok Wai Wong, Tom Gedeon, Alex Chong and Sebastian Khor

*School of Information Technology*
*Murdoch University*
*South St, Murdoch*
*Western Australia 6155*

**ABSTRACT:**

This paper examines the use of neuro-fuzzy techniques to perform intelligent data analysis. The advantages and disadvantages of using neural networks and fuzzy models in intelligent data analysis are investigated and reported. A comparison is made of neural networks, fuzzy rules extracted from data and fuzzy rules extracted from trained neural networks. A neuro-fuzzy data analysis model is also proposed. It is often observed that the data analysis cycle consists of three main steps: feature selection, model construction and interpretation of the model. It is shown that the use of the proposed neuro-fuzzy model can be beneficial in all the three steps. The proposed technique generates results as good as that of other techniques and possesses desirable features when compared to other techniques in the literature.

**KEYWORDS:** Intelligent data analysis, neuro-fuzzy, fuzzy rules extraction, feature selection

## 1.0 INTRODUCTION

In many engineering and business applications, data analysis plays an important role. The data analysis approach used must be able to provide a reasonable summary as well as an analysis of the data. There are two broad categories of data analysis; descriptive and inferential [1,2]. Intelligent data analysis systems are becoming popular especially with the use of neural networks, fuzzy systems and neuro-fuzzy systems. Fuzzy systems and neural networks are complementary techniques for designing an intelligent data analysis system, with its own advantages and disadvantages [3]. Neural networks are well known for their application to classification and function approximation problems. They have the ability to perform non-linear input and output mapping from training data, and are capable of generalisation by rejecting noise and generating results for input data that are new to the network. On the other hand, fuzzy systems have the ability to handle fuzzy information and can also handle non-linear functions. The major advantage of fuzzy system is the ability to express the ambiguity of knowledge in linguistic terms. The inspiration for neural networks and fuzzy system historically comes from the desire to produce artificial systems capable of sophisticated intelligent computation.

When using neural networks and fuzzy systems independently, each technique has its shortcomings that prevent it from being a "complete" solution for use as an intelligent data analysis system. For example, after a neural network is trained, it acts like a "black-box". A user will have difficulty in understanding the large number of weights involved. In addition, the effects on the output are unpredictable if some weights are modified. As for a fuzzy system, the setting up of the fuzzy rules can be tedious especially when a large number of input parameters is involved. Also, it does not have the ability to learn and adapt from the training data. To solve the problem of how to set up fuzzy rules, fuzzy rule extraction/induction algorithms can be used to obtain the rules from the training data [4,5,6]. However, the rules extracted may not have the best generalisation capability. Also, the rules generated may not cover the whole range of the universe of discourse. With these disadvantages, there is a need to integrate the two techniques.

There are many ways that the combination can be implemented [7]. For the Neural Fuzzy Networks technique, the intelligent data analysis system normally makes use of fuzzy methods to enhance the learning capabilities or performance of a neural network. In this case, after the neural network has learned the underlying function, it is still acting as a "black-box" with difficulties for humans in interpreting the data analysis model. When using a Cooperative Neuro-Fuzzy technique, a neural network is used to learn the underlying function and fuzzy rules are extracted from the trained neural network. After the fuzzy rules are extracted, then the neural network is not used any more. This way of combining neural networks and fuzzy systems is desirable for use as a model to design our intelligent data analysis system.

This paper acts as a comparison study to investigate the use of a Backpropagation Neural Network (BPNN), a fuzzy data analysis model using fuzzy rule extraction techniques, and our proposed neuro-fuzzy data analysis system. In most data analysis cycles, normally there are three main steps: feature selection, building the data analysis model and interpretation of the data analysis model. The proposed neuro-fuzzy technique can effectively handle these three steps easily and automatically. The comparison study also shows that this technique has many desirable features as compared to the others techniques used in this paper. Also, this paper shows that the proposed neuro-fuzzy approach can generate as good or better results than in the literature.

## 2.0 FEATURE SELECTION

In most intelligent data analysis systems, normally the first step is to perform feature selection, or input data mining. The main purpose of the feature selection is to identify the significant input variables in predicting the output [9]. This is important for most data mining and data analysis problems, as the available number of input variables may be very large. The significant inputs for some cases have a direct relationship with their correlation to the output. In some cases by pruning some input variables (irrelevant inputs), the results can be improved. In our intelligent data analysis system, the first step is to select the most contributing input variables for constructing the data analysis model. For our study, in order to confidently identify the significant input variables used to build the data analysis model, we take a look at four feature selection methods namely Garson [10], Milne [11], Gedeon and Wong [9, 12], and Fung, Wong and Crocker [13].

Garson [10] proposed the measure show in equation (1) for the proportional contribution of an input to a particular output. This is calculated as the fraction a single weight to a hidden neuron makes to all weights to that neuron, modulated by the weight connection to output. Then they divide by the sum of all such paths.

$$G_{ik} = \frac{\sum_{j=1}^{nh} \frac{w_{ij}}{\sum_{p=1}^{ni} w_{pj}} \bullet w_{jk}}{\sum_{q=1}^{ni} \left( \sum_{j=1}^{nh} \frac{w_{qj}}{w_{pj}} \bullet w_{qj} \right)} \quad (1)$$

A disadvantage of this approach is that during the summation process, positive and negative weights can cancel their contribution which leads to inconsistent results.

Milne [11] commented that the sign of the contribution is lost, and proposed equation (2)

$$M_{ik} = \frac{\sum_{j=1}^{nh} \frac{w_{ij}}{\sum_{p=1}^{ni} |w_{pj}|} \bullet w_{jk}}{\sum_{q=1}^{ni} \left( \left| \sum_{j=1}^{nh} \frac{w_{qj}}{\sum_{p=1}^{ni} |w_{pj}|} \bullet w_{qj} \right| \right)} \quad (2)$$

Similar to Garson, but the fraction uses the absolute value of the weights to that hidden neuron. Also divide by the sum of absolute values of all paths. The meaning of this sum is not intuitively clear.

Gedeon and Wong [9, 12] used the measure shown in equation (3) for the contribution of an input to a neuron in the hidden layer.

$$P_{ij} = \frac{|w_{ij}|}{\sum_{p=1}^{ni} |w_{pj}|} \quad (3)$$

This is the fraction of contribution the absolute value of weights makes to the sum of absolute values.

A measure $P_{jk}$ is defined for the contribution of a hidden neuron to an output neuron similar to the measure $P_{ij}$ used in equation (3) is shown in equation (4).

$$P_{jk} = \frac{|w_{jk}|}{\sum_{r=1}^{nh} |w_{rk}|} \quad (4)$$

The contribution of an input neuron to an output neuron [9] is then shown in equation (5).

$$Q_{ik} = \sum_{r=1}^{nh} (P_{ir} \times P_{rk}) \quad (5)$$

The benefit of this approach is that the magnitude of the contribution is disentangled from the sign of the contribution. The magnitude of contributions is significant in indicating whether an input is important, while the sign of the contribution is largely irrelevant in the decision to remove or retain an input, and is recoverable in any case from the raw data by simple statistical methods.

Fung, Wong and Crocker [13] identify the input variables without examining at the weights of the BPNN, but assuming that the trained weights should present some kind of sensitivity by the derivative of the output if the input is important in the model. They vary each input variables to their maximum and minimum and observe the derivatives of the change between the two. Finally the contribution of each input variable is calculated by equation (6).

$$C_k = \frac{T\Delta o|i_k}{\sum_{j=1}^{n} (T\Delta o|i_j)} * 100\% \quad (6)$$

This method is computationally cheaper than the other techniques.

## 3.0 BACKPROPOGATION NEURAL NETWORKS

A Backpropagation Neural Network (BPNN) is used in this study. BPNNs are the most widely used neural network systems and the most well known supervised learning technique used in intelligent data analysis systems [14]. Validation techniques [15,16,17] should be used to ensure the best generalization point to stop training. This is beyond the scope of this paper and therefore not discussed here.

## 4.0 FUZZY RULE EXTRACTION

The fuzzy rule extraction technique [6] investigated in this paper is an extension of the technique proposed by Sugeno and Yasukawa (SY) in [18] to produce IF-THEN rules. Given a set of training data, the technique first clusters the output space. Data points from each output cluster are projected back to each input dimension forming one-dimensional clusters. The clusters from different dimensions are then merged to form fuzzy rules. In [6], the technique was validated using artificially generated as well as well-known benchmark data sets. It was shown that the technique has reasonable accuracy using relatively few fuzzy rules.

The algorithm consists of 6 steps:

1. Perform Fuzzy c-Means clustering [19] on the output space. The algorithm iteratively searches for a a set of cluster centers that represent the structure of the data as best as possible by minimizing (8).

$$J_m(U,V) = \sum_{k=1}^{n}\sum_{i=1}^{c}(U_{ik})^m \| x_k - v_i \|^2,\ 1 \le m \le \infty \qquad (8)$$

where $J_m$(U,V) is the sum of squared error for the set of fuzzy clusters represented by the membership matrix $U$, and the associated set of cluster centers $V$. Here, $||x_k - v_i||^2$ represents the distance between the data $x_k$ and the cluster center $v_i$. At each iteration, the cluster centers are calculated using (9) and (10).

$$U_{ik} = \left( \sum_{j=1}^{c} \left( \frac{\| x_k - v_i \|}{\| x_k - v_j \|} \right)^{2/(m-1)} \right)^{-1} \forall i, \forall k \qquad (9) \text{ and } v_i = \frac{\sum_{k=1}^{n} (U_{ik})^m x_k}{\sum_{k=1}^{n} (U_{ik})^m} \qquad (10)$$

The optimal number of clusters in the data is determined by means of the FS index [20] as follows:

$$S(c) = \sum_{k=1}^{n}\sum_{i=1}^{c}(U_{ik})^m(\|x_k - v_i\|^2 - \|v_i - \bar{x}\|^2)\ \ 2<c<n \qquad (11)$$

The number of cluster, $c$, is determined so that $S(c)$ reaches a local minimum as $c$ increases. The clusters formed are fuzzy clusters in the sense that they are allowed to overlap with adjacent clusters as long as $S(c)$ is minimised.

2. For each fuzzy output cluster $B_i$ approximated, all the points belonging to the cluster are projected back to each of the input dimensions. For each dimension, fuzzy clustering is again applied to the projection of the points. In this step, the FS index (11) is used in conjunction with the merging index [21]:

$$P(v) = \sum_{j=1}^{n} e^{-4\left\| (v - x_j)/\frac{(v_i - v_j)}{2} \right\|^2} \qquad (12)$$

For each pair of cluster centers $v_i$ and $v_j$, the index (12) is merged if $p(v_m)$ is smaller than both $p(v_i)$ and $p(v_j)$, where $v_m$ is the middle point $(v_i + v_j)/2$.

3. The previous step results in multiple 1D fuzzy clusters in each input dimension. For each fuzzy cluster, a trapezoidal cluster is approximated. We refer the reader to [22] for a simple trapezoidal cluster approximation technique. The partition is converted to a Ruspini partition [23] for the convenience of the latter steps.

4. Each of the $n$ clusters $(C_{d1} - C_{dn})$ in the input dimension $d$, is a projection of the multi-dimensional input cluster to that input dimension. Next the clusters from individual dimensions are combined to form the multi-dimensional input cluster. The merging process involves the use of a threshold $t$ which governs the degree of sparseness in the rule-base to be generated. In general, the higher the threshold, the fewer rules are generated.

The cluster in the multi-dimensional space is determined to be the region where the number of projected points in the region exceeds $t$. A point $p$ is contained in the cluster $C_i$ if $\mu_{Ci}(p) > \mu_{Cj}(p)$ for all $j \neq i$. The process has three steps:

a. Find one of the multi-dimensional clusters $C$ where the number of points that falls into its projection exceeds the threshold $t$ using the following algorithm:

```
PROCEDURE find_MD_cluster
Let Ui be the set of one-dimensional clusters in dimension i
Let mdCluster = [ ]
for i = 1 to k
    for each unit u ∈ Ui
       utemp = mdCluster x u
       if utemp is dense
         denseunit = utemp
         break
       end if
    end for
end for
```

b. Remove all data points that are contained in the cluster $C$ approximated.
c. Repeat steps 1 – 2 until no more clusters can be found.

5. For each of the multi-dimensional clusters identified, a rule can be formed. For example, if a multi-dimensional cluster is formed with $[C_{11}, C_{23}, C_{34}]$ for the points projected from output cluster $B_i$, we obtain the following rule:
   *If $x_1$ is $C_{11}$ and $x_2$ is $C_{23}$ and $x_3$ is $C_{34}$ then y is $B_i$*
   Where $C_{dn}$ is the $n^{th}$ cluster identified at input dimension $d$.

6. The completed fuzzy rule-base then goes through a parameter identification process where each trapezoidal cluster in the input and output space is adjusted to improve the overall performance. The parameter identification is described in [18]. An alternative technique is proposed in [23]. These details are omitted here as they only fine tune the rules.

## 5.0 NEURO-FUZZY

The objective of this proposed technique is to set up an intelligent data analysis system that is comprehensible by the user. The rules deduced should also describe the underlying function of the training data by excluding noise or outliers. From the previously described techniques, each has its strong and weak points. The BPNN has the ability to generalised and eliminate noise. However, once the network is trained, it is difficult to understand how the system operates. The user cannot modify the behaviour of the model. The proposed neuro-fuzzy intelligent data analysis system combines the two approaches. Establishment of the system is basically divided into two parts. The first part involves the training of a generalised BPNN based on backpropagation learning algorithm. The second part involves extracting fuzzy rules to explain the underlying function of the trained neural network.

This suggests creating the intelligent data analysis system as follows. After the number of memberships required for the fuzzy system is determined using one of the feature selection methods described earlier, input variables for all possible memberships are generated and applied to the trained BPNN. The outputs generated cover the universe of discourse of the sample space. The set of generated input variables with their corresponding outputs from the BPNN model are now used as the training data for a fuzzy rule extraction system. As this data set also describes the generalisation function underlying the BPNN data analysis model, the fuzzy rules produced will encapsulate the required knowledge. This is the model construction phase. After the fuzzy rules have been extracted, the BPNN is not used any more for prediction. Any new inputs will be input to the fuzzy system for prediction. The fuzzy rule base is used for interpretation of any particular output via the specific fuzzy rules which fired to create the output.

## 6.0 CASE STUDY

The dataset used in this comparison study was obtained from the Statistics library maintained by Carnegie Mellon University. This dataset concerns the housing values in the suburbs of Boston. There are 13 input variables used to determine the housing price. The input and output variables are tabulated in Table 1. We have randomly separated the whole data set of 506 data points into two sets: for training (253 data points) which is divided into two sets: training set and validation set, and testing set (253 data points).

Table 1: Attributes of the housing dataset

| Attributes | Descriptions |
|---|---|
| (1) CRIM | per capita crime rate by town |
| (2) ZN | proportion of residential land zoned for lots over 25,000 sq.ft. |
| (3) INDUS | proportion of non-retail business acres per town |
| (4) CHAS | Charles River dummy variable (= 1 if tract bounds river; 0 otherwise) |
| (5) NOX | nitric oxides concentration (parts per 10 million) |
| (6) RM | average number of rooms per dwelling |
| (7) AGE | proportion of owner-occupied units built prior to 1940 |
| (8) DIS | weighted distances to five Boston employment centres |
| (9) RAD | index of accessibility to radial highways |
| (10) TAX | full-value property-tax rate per $10,000 |
| (11) PTRATIO | pupil-teacher ratio by town |
| (12) B | 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town |
| (13) LSTAT | % lower status of the population |
| (14) MDEV | Median value of owner-occupied homes in $1000's |

After the feature selection, the significant input variables that can be used to predict the median value of the house are determined. The four methods (Garson [10], Milne [11], Gedeon and Wong [9, 12], and Fung, Wong and Crocker [13]) described in Section 2.0 are used, and results are shown in Table 2. From Table 2, we found that inputs (6) RM, (8) DIS, (10) TAX, and (13) LSTAT to be the more important features from the four methods. To justify that the four inputs selected are correct, we have established comparison models: one with all input variable and the other with only the four input variables. The results are tabulated in Table 3.

Table 2: Feature Selection Results

| Methods | Sort according from the most significant inputs | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fung, Wong and Crocker | **10** | **8** | **13** | **6** | 5 | 1 | 11 | 9 | 2 | 7 | 12 | 3 | 4 |
| Gedeon and Wong | **6** | **13** | **8** | **10** | 9 | 11 | 1 | 3 | 5 | 12 | 7 | 4 | 2 |
| Milne | **13** | **6** | **8** | 1 | 5 | 11 | 3 | 12 | 9 | 10 | 2 | 7 | 4 |
| Garson | **6** | 1 | 3 | **10** | 4 | 7 | 2 | 11 | 12 | **8** | 5 | **13** | 9 |

Table 3: Comparison of testing results for feature selection

| Data analysis model | Correlation R-Square | RMSE |
|---|---|---|
| BPNN (using all 14 inputs) | 0.80 | 4.24 |
| BPNN (using only 4 inputs) | 0.80 | 4.02 |
| Fuzzy rule extraction from data (using all 14 inputs) | 0.45 | 6.78 |
| Fuzzy rule extraction from data (using only 4 inputs) | 0.56 | 6.10 |

Table 4: Comparison of different intelligent data analysis model

| Intelligent data analysis systems | Correlation R-Square | RMSE |
|---|---|---|
| BPNN | 0.80 | 4.02 |
| Fuzzy rule extraction from data | 0.56 | 6.10 |
| Neuro-Fuzzy from BPNN | 0.79 | 3.46 |

From Table 3, we can see that the feature selection techniques have confidently selected the significant inputs for predicting the housing prices. We also notice that the neural network is better than the fuzzy rule extraction from the data.

After the 4 input variables have been determined, the comparison will only make use of the reduced data set. Table 4 shows the comparison results of the different intelligent data analysis systems. In both tables, the correlation and RMSE are between expected and predicted values.

For the BPNN, the best neural network configuration found is 4-8-1. The training is performed using 134 training data and 119 validation data. For the fuzzy rule extraction technique, the number of fuzzy rules extracted is 21. For the neuro-fuzzy intelligent data analysis system, the results are quite similar to that of the BPNN. It therefore shows that the fuzzy extraction technique can extract the underlying function from the trained BPNN. The number of fuzzy rules extracted by the neuro-fuzzy technique is 17. By combining the BPNN and fuzzy rule extraction techniques, we have incorporated the advantages of the neural network and fuzzy modelling. It can be observed that the proposed neuro-fuzzy has better prediction accuracy than the fuzzy rule extraction system. It is also noted that the neuro-fuzzy system uses fewer fuzzy rules. The main reason is that the BPNN has generalised from the data before generating information for the fuzzy rule extraction technique used in extracting the underlying function. The

prediction accuracy of the neuro-fuzzy is essentially the same as the BPNN, and has the advantage of presenting only 17 fuzzy rules for the user to understand the data analysis model.

## 7.0 CONCLUSION

A neuro-fuzzy intelligent data analysis system is proposed in this paper, which integrates the main features of a BPNN and a fuzzy rule extraction system. The strong points of the two systems are combined into one intelligent data analysis model. The final system is capable of generalising from the available training samples, to perform interpretation from new data and to provide users with a set of human understandable fuzzy rules. The users may examine the rule base and perform necessary modification or add new rules to the system based on past experience and knowledge. The comparison study has shown that this proposed approach provides good prediction results.

## REFERENCES


[1] W. Mendenhall and T. Sincich, T. *Statistics for Engineering and the Sciences*, 3rd Edition, Dellen Publishing Company, 1992.

[2] M.C. Phipps and M.P. Quine. *A Primer of Statistics: Data Analysis, Probability, Inference*, 3rd Edition,Prentice Hall, 1998.

[3] C.C. Fung and K.W. Wong "Establishing a Generalised Fuzzy Interpretation System Using Artificial Neural Network," in *Proceeding of the International Conference on Computational Intelligence and Multimedia Applications,* Melbourne, pp. 330-335, 1998.

[4] C.C. Fung, K.W. Wong and P.M. Wong. "A Self-generating Fuzzy Rules Inference Systems for Petrophysical Properties Prediction," in *Proceedings of IEEE International Conference on Intelligent Processing Systems*, pp. 205-208, 1997.

[5] S. Guillaume. "Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review," *IEEE Transactions on Fuzzy Systems*, 9(3): pp. 426-443, 2001.

[6] A. Chong, T.D. Gedeon and L.T. Kóczy. "A Projection Based Method for Sparse Fuzzy System generation", in *Proceedings of 2nd WSEAS International Conference on Scientific Computation and Soft Computing*, pp. 321-325, 2002.

[7] D. Nauck. "Beyond Neuro-Fuzzy: Perspectives and Directions", *Proceedings of the Third European Congress on Intelligent Techniques and Soft Computing*, pp. 1159-1164, 1995.

[8] C. Chong, T.D. Gedeon and K.W. Wong. "Extending the Decision Accuracy of a Bioinformatics System," in *Proceedings of 3rd Western Australia Workshop on Information Systems Research (WAWISR'00)*, Paper 140, 2000.

[9] T.D. Gedeon. "Data mining of inputs: analysing magnitude and functional measure," *International Journal of Neural Systems*, vol. 8 (2) pp. 209-218, 1997.

[10] G.D. Garson. "Interpreting Neural Network Connection Weights," *AI Expert*, pp. 47-51, 1991.

[11] LK Milne. "Feature Selection Using Neural Networks with Contribution Measures," in *Proceedings of the Australian Conference on Artificial Intelligence (AI'95)*, Canberra, 1995.

[12] P.M. Wong, T.D. Gedeon, and I.J. Taggart. "An Improved Technique in Porosity Prediction: A Neural Network Approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, n. 4, pp. 971-980, 1995.

[13] C.C. Fung, K.W. Wong, and H. Crocker. "Determining Input Contributions for a Neural Network Based Porosity Prediction Model," in *Proceedings of Eighth Australian Conference on Neural Network (ACNN'97)*, pp. 35 – 39, 1997.

[14] D.E. Rumelhart, G.E. Hinton and R.J. Williams. "Learning Internal Representation by Error Propagation" in *Parallel Distributed Processing*, vol. 1, MIT Press, pp. 318-362, 1986.

[15] C. Wang, S.S. Venkatesh and J.S. Judd. "Optimal Stopping and Effective Machine Complexity in Learning", *Advances in Neural Information Processing Systems*, 6, pp. 303-310, 1994.

[16] A. Weigend, D. Rumelhart and B. Huberman. "Generalisation by Weight Elimination with Application to Forecasting", *Advances in Neural Information Processing Systems*, 1991.

[17] S. Lawrence, C.L. Giles and A.C. Tsoi,. "What Size Neural Network Gives Optimal Generalisation? Convergence Properties of Backpropagation", *Technical Report UMIACS-TR-96-22 & CS-TR-3617*, Institute for Advanced Computer Studies, University of Maryland, 1996.

[18] M. Sugeno and T. Yasukawa. "A fuzzy-logic-based approach to qualitative modelling," *IEEE Transactions on Fuzzy Systems*, 1(1), pp. 7-31, 1993.

[19] J.C. Bezdek. *Pattern Reconition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.

[20] Y. Fukuyama and M. Sugeno. "A new method of choosing the number of clusters for fuzzy c-means method," in *Proceedings of the 5th Fuzzy Systems Symposium*, pp. 247-250, 1989.

[21] A. Chong, T.D. Gedeon, and L.T. Koczy. "A Hybrid Approach for Solving the Cluster Validity Problem". submitted to *14th International Conference on Digital Signal Processing*, pp. 1207-1210, 2002

[22] D. Tikk, T.D. Gedeon, L.T. Koczy, and G. Biro. "Improvements and critique on Sugeno and Yasukawa's qualitative modelling," *IEEE Transactions on Fuzzy Systems*, 2002.

[23] E.H. Ruspini. "A new approach to clustering," *Information and Control*, 15, pp. 22-32, 1969.