Focus

Soft Computing 7 (2003) 434 – 438 © Springer-Verlag 2003
DOI 10.1007/s00500-002-0232-4

Rainfall prediction model using soft computing technique

K. W. Wong, P. M. Wong, T. D. Gedeon, C. C. Fung

Abstract Rainfall prediction in this paper is a spatial interpolation problem that makes use of the daily rainfall information to predict volume of rainfall at unknown locations within area covered by existing observations. This paper proposed the use of self-organising map (SOM), backpropagation neural networks (BPNN) and fuzzy rule systems to perform rainfall spatial interpolation based on local method. The SOM is first used to separate the whole data space into some local surface automatically without any knowledge from the analyst. In each sub-surface, the complexity of the whole data space is reduced to something more homogeneous. After classification, BPNNs are then use to learn the generalization characteristics from the data within each cluster. Fuzzy rules for each cluster are then extracted. The fuzzy rule base is then used for rainfall prediction. This method is used to compare with an established method, which uses radial basis function networks and orographic effect. Results show that this method could provide similar results from the established method. However, this method has the advantage of allowing analyst to understand and interact with the model using fuzzy rules.

Keywords Self-organising map, Backpropagation neural networks, Fuzzy system, Spatial interpolation, Geographic information system

1 Introduction

Rainfall prediction is one of the application areas in spatial interpolation, which is an important feature of a Geographic Information System that uses procedure to estimate values at unknown locations within the area covered by existing observations [1]. In this paper, it is used to estimate volume of rainfall at unknown locations within the area, by establishing an interpolation model based on the rainfall observations in the region. All spatial interpolation techniques can be grouped into global and local methods [1]. In a global method, all the information available is used to estimate an unknown value, while local methods only use a sample of the information for estimation. In a global method, trend surface analysis is normally performed. The equation that can be used to estimate values at other points using a third-order trend surface is:

$$Z = b_0 + b_1X + b_2Y + b_3X^2 + b_4XY + b_5Y^2 + b_6X^3 + b_7X^2Y + b_8XY^2 + b_9Y^3 \quad (1)$$

where the b coefficients are estimated from the available information points.

Equation (1) can also be written as:

$$Z = f(X, Y) \quad (2)$$

As for a local method, the spatial interpolation of the value Z_i in the ith surface is:

$$Z_i = f(X_i, Y_i) \quad (3)$$

This trend surface analysis could be considered as a subset of polynomial regression [2].

Artificial Neural Networks (ANNs) are becoming popular for spatial data analysis [3, 4]. ANN analysis is quite similar to statistical approaches in that both have learning algorithm to help them realise the data analysis model. However, an ANN has the advantages of being robust with the ability to handle large amounts of data. Novice users can also easily understand the practical use of an ANN. An ANN also has the ability to handle very complex functions [5]. The main limitation of using ANN is that the data analysis model built may not be able to be interpreted. Fuzzy logic is also becoming popular in dealing with data analysis problems that are normally handled by statistical approaches or ANNs [6]. However, conventional fuzzy systems do not have any learning algorithm to build the analysis model. Rather, they make use of human knowledge, past experience or detailed analysis of the available data by other means in order to build the fuzzy rules for the data analysis. The advantages of using fuzzy systems are the ability to interpret the analysis model built and to handle vagueness and uncertainty in the data. The data analysis model can also be changed easily by modifying the fuzzy rule base. The major limitation is the difficulty in building the fuzzy rules due to lack of learning capability.

ANNs and fuzzy logic are complementary technologies in designing an intelligent data analysis approach [7]. That suggests combining the two [8]. However, there are many ways that the combination can be implemented. Table 1

K. W. Wong (✉), T. D. Gedeon
School of Information Technology, Murdoch University,
South St, Murdoch, Western Australia 6150
e-mail: kwong@murdoch.edu.au

P. M. Wong
School of Petroleum Engineering,
University of New South Wales, Sydney, NSW 2052, Australia

C. C. Fung
School of Electrical and Computer Engineering,
Curtin University of Technology,
Kent St, Bentley, Western Australia 6102

Table 1. Different ways to combine ANN and fuzzy logic

Techniques	Description
Fuzzy Neural Networks	Use fuzzy methods to enhance the learning capabilities or performance of ANN
Concurrent Neuro-Fuzzy	ANN and fuzzy systems work together on the same task without any influence on each other
Cooperative Neuro-Fuzzy	Use ANN to extract rules and then it is not used any more
Hybrid Neuro-Fuzzy	ANN and fuzzy are combined into one homogeneous architecture

shows the different ways that ANN and fuzzy systems can work together. It is important to observe the characteristics under each class so as to determine the appropriate technique that the analyst will be comfortable with.

The Cooperative Neuro-Fuzzy technique is selected as the more appropriate technique to be used in this application. The reasons are as follows. As the backpropagtion neural network (BPNN) can generalize from the data through some learning algorithm, the spatial interpolation function could be realised automatically. This will also enable the fuzzy rules to cover the whole universe of discourse, so that they can be used to approximate data that are not present in the training set. As fuzzy rules are closer to human reasoning, the analyst could understand how the rainfall model performs prediction. If necessary, the analyst could also make use of his/her knowledge to modify the prediction model. The prime objective of this paper is to examine the use of Cooperative Neuro-Fuzzy technique in building a rainfall prediction model. A case study uses data collected on the 8th May 1996 in Switzerland is used to test this soft computing technique. Results are compared with an established ANN technique [3], and shown that they have similar results. However, with the use of this soft computing technique, the analyst could understand the rainfall prediction model by examining the fuzzy rule base, so as to modify and interact with the model.

2 Identifying local surface

For local spatial interpolation, the first step is to classify the available data into different classes, so that the data are split into homogeneous sub-populations. Self-organizing Map (SOM) [9, 10] is used to divide the data into sub-populations and hopefully reduce the complexity of the whole data space to something more homogeneous. The objective in this step is to make use of an unsupervised learning algorithm to sub-divide the whole population. The SOM is selected for this purpose mainly because it is a fast, easy and reliable unsupervised clustering technique.

SOM is designed with the intention to closely simulate the various organisations found in various brain structures and has a close relationship to brain maps. Its main feature is the ability to visualise high dimensional input spaces onto a smaller dimensional display, usually two-dimensional as shown in Fig. 1. In this discussion, only two-dimensional arrays will be of interest. Let the input data space $\mathscr{R}^n$ be mapped by the SOM onto a two-dimensional array with i nodes. Associated with each i node is a parametric reference vector $m_i = [\mu_1, \mu_2, \ldots, \mu_{in}]^T \in \mathscr{R}^n$, where μ_{ij} is the connection weight between node i and input j. Therefore, the input data space $\mathscr{R}^n$ consisting of input vectors $X = [x_1, x_2, \ldots, x_n]^T$, i.e. $X \in \mathscr{R}^n$, can be visualized as being connected to all nodes in parallel via a scalar weight μ_{ij}. The aim of the learning is to map all the n input vectors X_n onto m_i by adjusting weights μ_{ij} such that the SOM gives the best match response locations.

SOM can also be said to be a nonlinear projection of the probability density function $p(X)$ of the high dimensional input vector space onto the two-dimensional display map. Normally, to find the best matching node i, the input vector X is compared to all reference vector m_i by searching for the smallest Euclidean distance $||X - m_i||$, indexed by c, i.e. $||x - m_c|| = \min_i ||x - m_i||$.

During the learning process the node that best matches the input vector X is allowed to learn. Those nodes that are close to the node up to a certain distance will also be allowed to learn. The learning process is expressed as:

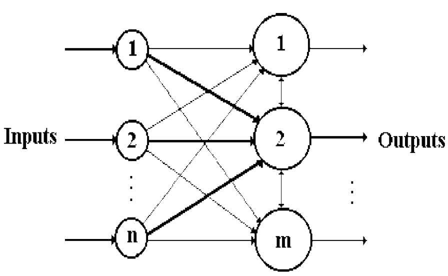
$$m_i(t+1) = m_i(t) + h_{ci}(t)[X(t) - m_i(t)] \quad (4)$$

where t is a discrete time coordinate and $h_{ci}(t)$ is the neighbourhood function.

After the learning process has converged, the map will display the probability density function $p(X)$ that best describes all the input vectors. At the end of the learning process, an average quantisation error of the map will be generated to indicate how well the map matches the entire input vectors X_n. The average quantisation error is defined as:

$$E = \int ||X - m_c||^2 p(X) \mathrm{d}X \quad (5)$$

After the 2-dimensional map has been trained, the reference vectors that were used in the nodes of the map can be also obtained. In rainfall prediction, the reference vector will be the node centre and consists of the input variables



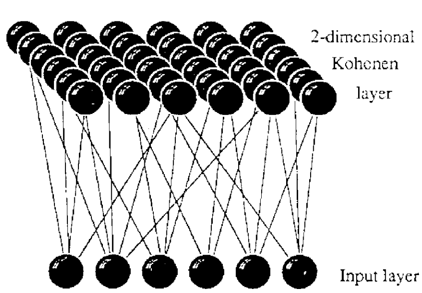


Fig. 1. Visualisation for self organising map

(x, y) and the output variable (z). As we like the clusters to be formed to facilitate the concept used in Euclidean interpolation, we propose here to construct the clustering boundaries based on the output reference vector of the nodes. The rule of thumb for deciding on the clustering boundaries is to examine the distance measure between the neighbouring reference values. If the distance measure between the present reference node and the neighbouring nodes is high, that suggests another cluster. Figures 2 and 3 shows an example of drawing the boundaries.

3 Rainfall prediction model

After the local surfaces have been identified by the SOM, the set of available training data are then sub-divided into the local surfaces. BPNNs are trained in each sub-surface to predict only data belonging to the same sub-surface. Therefore, if the SOM identified c clusters, then c BPNNs need to be trained. When a BPNN [11] is used in rainfall prediction, the rainfall observations obtained from the neighbouring are used as the training data, thus it is a supervised learning technique. The input neurons of the BPNN in this case correspond to the x and y position coordinates, and the output neuron is assigned to the rainfall measurements (z) on which we want to perform spatial interpolation. The BPNN has a number of layers. The input layer consists of all the input neurons and the output layer (just the output neuron). There are also one or more hidden layers. All the neurons in each layer are connected to all the neurons in next layer with the connection between two neurons in different layers represented by a weight factor. After the BPNN has learned and generalised from the training data, it is then used to construct the fuzzy rules bases as shown in Fig. 4.

As all the BPNNs have generalized from the training data, the next step is to extract the knowledge learned by the BPNNs. In this case, it is the same as the previous section; we will have to extract c fuzzy rule bases. The

	0	1	2	3	4	5	6	7	8	9
0	49.07	68.49	101.39	146.15	195.26	251.58	278.66	346.97	305.80	248.81
1	68.98	119.63	127.22	132.80	202.07	216.55	313.67	325.59	285.68	216.18
2	170.97	147.21	176.29	198.65	147.32	101.23	264.75	303.90	231.52	146.79
3	125.29	155.71	156.49	138.93	145.27	221.58	351.26	352.78	242.52	117.42
4	126.69	125.32	137.97	110.82	122.38	250.33	321.66	348.32	166.78	120.25
5	130.86	119.28	118.82	77.96	111.27	164.85	282.00	365.54	340.93	171.91
6	124.37	131.55	80.91	84.45	61.10	143.53	207.36	356.25	265.77	197.01
7	184.45	288.04	179.71	64.20	90.60	142.81	274.69	452.64	302.09	165.12
8	271.93	311.74	283.21	161.43	81.58	127.68	116.04	346.14	291.78	157.07
9	205.36	270.88	226.25	56.87	67.34	90.67	107.63	262.06	318.77	268.53

Fig. 2. The SOM vector map for the output (z)

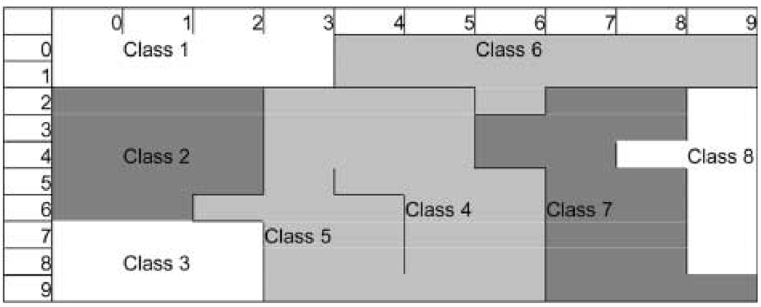


Fig. 3. The cluster boundaries for on the SOM 2-dimensional map based on output (z)

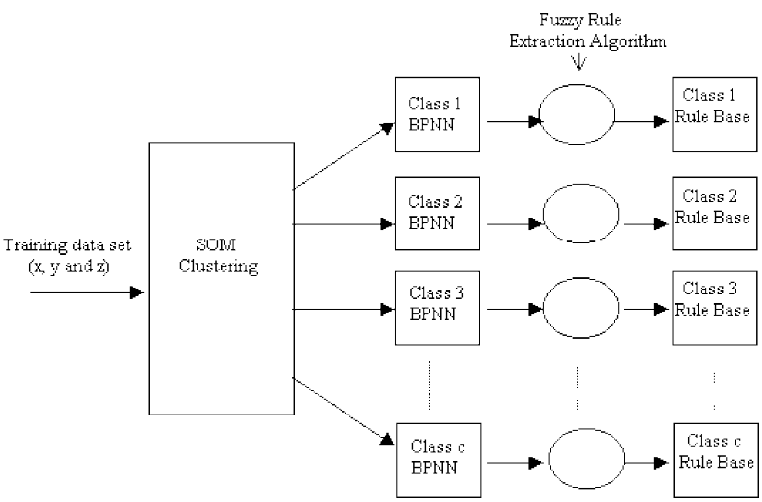


Fig. 4. The neural fuzzy spatial interpolation model

following algorithm outlines the steps in extracting the fuzzy linguistic rules for one BPNN.

As we have to extract fuzzy rules that can cover the whole universe of discourse in order to cover the whole sample space as seen by the BPNN, for T membership functions or linguistics terms, we would have T^2 fuzzy rules as we have only two variables (x, y) in this case. We randomly generate input variables that could cover all the possible input space as seen by the BPNN and input them into the BPNN to obtain the rainfall measurements predicted by the BPNN.

For the two inputs (x, y), the BPNN generates input (x, y)-output (z) data pairs with n patterns:

$$(x^1, y^1; z^1)$$
$$\vdots$$
$$(x^n, y^n; z^n)$$

The number of linguistics terms T used in this fuzzy rule extraction has to be the same as the predetermined one when generating output from the BPNN. The distribution of the membership functions in each dimension of the domain in this case is evenly distributed. For ease of interpolation and computational simplicity, the shape of the membership functions used in this rule extraction technique is a triangular fuzzy member as shown in Fig. 5.

After the fuzzy regions and membership functions have been distributed, the available input-output pairs will be mapped. If the value cuts on more than one membership function, the one with the maximum membership grade will be assigned to the value. For rule R_i, the membership function A_x^i for input x_i is taken as the A_x^i with maximum alpha cut. The process can then be illustrated by the following equation:

$$R^i \Rightarrow [x^i(A_x^i, \max), y^i(A_{y,}^i \max) : z^i(B^i, \max)] \quad (6)$$

After all the input-output values have been assigned to a fuzzy linguistic label, Mamdani type fuzzy rules are then formed [12] as follow:

$$R^i : \text{if } x \text{ is } A_x^i \quad \text{and} \quad y \text{ is } A_y^i \text{ then } z \text{ is } B^i \quad (7)$$

After the fuzzy rules base corresponding to the BPNN for a class have been constructed, the BPNN is not used anymore when performing spatial interpolation. With this set of fuzzy rules, a human analyst can now examine the behaviour of the interpolation. Changes and modification can then be performed if necessary. The fuzzy rules extracted can also handle fuzziness in the data and thus may improve the performance of the spatial interpolation.

4 Case study

In this case study, the data available from the AI-GEO-STATS mailing list in Italy [13] is used. The data is collected on 8th May 1996 in Switzerland. 100 data locations are used as the training data and the other 367 locations data are then used to verify the prediction accuracy of the established spatial interpolation model. The two input variables used in this case is the 2D coordinate position (x, y); and the output used is the rainfall measurements (z). The digital elevation model (DEM) (v) is also available but was not used by this technique. The 100 training data points are inputs to the SOM for unsupervised clustering. After performing the cluster boundaries determination, the classes are formed.

Before being inputs into individual BPNNs, the data needs to be normalized between 0 and 1. Linear normalization is used with maximum and minimum values unique to the class. In this case, the SOM identified a total of 8 classes or sub-surfaces. After the data has been normalized, 8 BPNNs are trained to handle their own sub-populations. After examining the maximum and minimum value of each class, the appropriate number of membership function used is determined to be 7. In this case, the number of fuzzy rules extracted for each BPNN (each class) is 49, i.e. 7^2. Part of the fuzzy rules used in class 1 are shown in Fig. 6, where EL is extremely low, VL is very low, L is low, M is middle, H is high, VH is very high, and EH is extremely high. With the distribution information for each linguistic term, the analyst can easily understand the set of fuzzy rules and understand how the prediction is performed.

This gives a summary of the technique used in [3] abbreviated as ANN technique, which is also used in this paper to make comparison with the results generated by the proposed soft computing technique. This ANN technique uses a divide-and-conquer approach in dividing the whole data space into four sub-regions for local spatial interpolation method. In ANN technique, they used different methods for different sub-surfaces. For two larger surfaces, they have used radial basis function (RBF) networks [14] to perform rainfall prediction. The other two smaller sub-surfaces, they have used a simple linear regression model to predict the rainfall. In their approach, they have made three assumptions. Firstly, they assumed that rainfall might have a different pattern in different areas. Secondly, in smaller areas, rainfall pattern is normally smooth and continuous. Thirdly, there exists orographic effect.

The minimum, maximum, mean, median and standard deviation of the 367 observed data and the interpolated

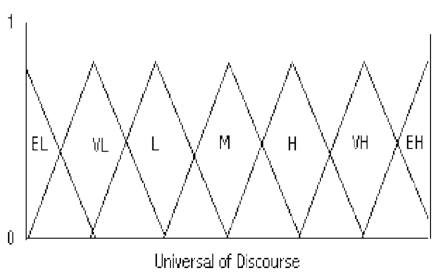


Fig. 5. Fuzzy membership used in x, y and z

```
If x = EL and y = EL then z = VL
If x = EL and y = VL then z = VL
If x = EL and y = L then z = VL
If x = EL and y = M then z = VH
If x = VL and y = EL then z = H
If x = VL and y = VL then z = H
If x = VL and y = M then z = M
If x = VL and y = EH then z = VL
If x = L and y = EL then z = VH
```

Fig. 6. Extract of a few fuzzy rules used to predict class 1 rainfall

Table 2. Comparison between observed and predicted rainfall

	Observed Z	Soft computing predicted Z	ANN predicted Z
Min	0	12	3
Max	517	467	522
Mean	185	194	182
Median	162	163	148
Standard deviation	111	110	112

Table 3. Error measures between the predicted and observed rainfall

Error measures	Soft computing technique	ANN technique
MAE	53.86	55.9
RMSE	72.95	78.65
Correlation measure	0.784	0.75
Relative error	0.31	0.46

data are tabulated in Table 2. From Table 2, the statistical comparisons of the results generated from the two techniques are quite comparable. This shows that the proposed soft computing techniques can be a valuable alternative for used in rainfall prediction. The relative mean absolute error (MAE), root mean square error (RMSE), the correlation measure, and the relative error between the predicted and observed rainfall are shown in Table 3. From Table 3, the different error measures used to verify the predicted and the observed rainfall measurements shows that this soft computing technique performs as good as the ANN technique. Two points need to be highlighted with this soft computing technique. Firstly, this method does not require any assumptions in dividing the whole surface into sub-surfaces; however, assumptions are made in the ANN method. SOM basically perform the clustering automatically. Secondly, this method generates a few sets of fuzzy rules to represent the knowledge of the BPNNs. This allows better human interaction and understanding into the rainfall prediction model.

5
Conclusion

In this paper, a soft computing technique that uses ANN and Fuzzy Logic has been used to establish a rainfall prediction model. This technique uses SOM to divide the data into sub-population and hopefully reduce the complexity of the whole data space to something more homogeneous. After the classification boundaries have been identified, the whole training data set is then sub-divided into the respective classes. BPNNs corresponding to each individual class are then trained using the cross-validation approach. After all the BPNNs have been trained, fuzzy rule bases for each class are then constructed. The case study has shown that this method can produce comparable results as those generated from the ANN technique. The advantages of using this technique are as follows. First it makes use of the robustness and learning ability of the ANN to sub-divide and generalize from training data. After which, the learned underlying function is then translated into fuzzy rules. With the use of fuzzy rules, the interpretability and the ability of handling vagueness and uncertainty has enhanced the interpolation model. Most important of all, this technique put forward a self-learning and self-explained rainfall prediction technique. The next phase of this research will emphasize examining the human understandable fuzzy rules in improving the prediction results.

References

1. **Burrough P, McDonnell R** (1998) Principles of geographical information systems. Oxford University Press, New York
2. **Myers RH** (1990) Classical and Modern Regression with Applications. PWD-Kent Publishing, Boston
3. **Lee S, Cho S, Wong PM** (1998) Rainfall prediction using artificial neural network. J Geographic Information and Decision Anal **2**(2): 254–264
4. **Friedman JH** (1994) An Overview of Predictive Learning and Function Approximation. From Statistics to Neural Networks: Theory and Pattern Recognition Applications, Berlin, Springer-Verlag, pp. 1–61
5. **Cherkassky V, Friedman JH, Wechsler H** (1994) From Statistics to Neural Networks: Theory and Pattern Recognition Applications. Springer-Verlag, Berlin
6. **Kosko B** (1997) Fuzzy Engineering. Prentice-Hall, New Jersey
7. **Williams T** (1994) Special report: bringing fuzzy logic and neural computing together, Computer Design: 69–84
8. **Nauck D** (1995) Beyond neuro-fuzzy: perspectives and directories, Proc 3rd European Congress on Intelligent Techniques and Soft Computing. Aachen, pp. 1159–1164
9. **Kohonen T** (1990) The self-organising map. Proc IEEE **78**(9): 1464–1480
10. **Kohonen T** (1995) Self-organising map, Springer Series in Information Sciences, Vol. 30, Berlin, Springer-Verlag
11. **Rumelhart DE, Hinton GE, Williams RJ** (1986) Learning internal representation by error propagation, Parallel Distributed Processing **1**: 318–362
12. **Mamdani EH, Assilian S** (1975) An experiment in linguistic synthesis with a fuzzy logic controller, Int J Man-Machine Studies **6**: 1–13
13. **Dubois G** (1998) Spatial Interpolation Comparison 97. ftp://ftp.geog.uwo.ca/SIC97/intro.html
14. **Park J, Sandberg I** Universal approximation using radial basis function networks, Neural Comput **3**(2): 246–257