利用模糊類神經網路解
型態分類的方法

張介仁[1] 王乃堅[2]

1. 國立宜蘭技術學院電子系講師
2. 國立台灣科技大學電機系副教授

摘 要

在資訊或生物科技工程上，分類問題一直是一個重要的研究方向，更是資料探勘領域上一們基礎的研究。類神經模糊規則架構常被運用在分類系統的研究上，本論文是以模糊量測（Fuzzy measures）為基礎，從資料中自動建立模糊規則庫及可調變的類神經結構並應用於分類系統上，並以範例加以實驗測試以證明此演算法的正確性。

本文中所提出的方法是利用輸入所有的訓練資料，並記錄下各類別每一個輸入變數的最小和最大值來建立包圍類別區域的超立方體作為類神經結構的第一層。接著測試這些超立方體之間重疊情形，我們分別定義了模糊量測、資訊提供度及資料分離度並建立一結構調變參數，用以判別此重疊區域是否需要再被更細的分割，而分割後的模糊規則庫即成為類神經結構的第二層。 依此類推，可建立多迴圈的類神經模糊架構，直到結構調變參數太小。此方法可以合理的使用訓練資料以建立有效的分類器，並可依照資料的不同來簡化計算的複雜度，已達快速分類之目的。

關鍵詞：分類器、模糊量測、模糊類神經網路、資料探勘

A Method of Fuzzy-Neural Network to Pattern Classification

Jieh-Ren Chang[1] and Nai-Jian Wang[2]

1. Lecturer, Department of Electronics Engineering, National Ilan Institute of Technology
2. Associate Professor, Department of Electric Engineering, National Taiwan Universaty of Technology

Abstract

The pattern classification is an important issue on Information technology and biological engineering. It is also a key element to data mining research. Recently, Fuzzy-Neural network system is used in many pattern classifiers. In this paper, a new method is proposed for setting a variable Fuzzy-Neural network structure directly from numerical data. It provides several examples of operation that demonstrate the strong qualities of this method.

We propose a variable fuzzy-neural structure network, which constructs by 3 layers for pattern classification. The first layer is for data input, and third layer is for output decision. We make the unit of second layer of network, which is set by each activation fuzzy hyper-box for each class. The fuzzy hyper-box is an n-dimensional box defined by a min point and max point with a corresponding membership function. Then, we test the condition of overlap of these hypercube by defined tuning-structure parameter, which is made by fuzzy measure, information support degree and data separation degree. We decide how many loops at most in the second layer of the network should be rebuild again. We also create a feedback node in third layer to decide the parameter value of each unit in second layer. We can generate a high efficiency classifier by this dynamic neural-fuzzy network structure using sufficient information of all training data. We also decrease the complexity of classification computation according to different test data.

Key Words：pattern classification, variable neural-fuzzy network structure, data mining, fuzzy measure

I. Introduction

Pattern classification is a key component to many engineering, such as radar, seismic, control, sonar, bio-information and diagnostic application. In case of significant computer progress, artificial intelligent pattern recognition still faces continuous big challenge from human recognition. Humans always can collect the knowledge from the uncertain or ambiguous data. So, it seems be solved more efficiently by human in classification problem which still can't be dealt perfectly in computer. Many methods still are proposed to improve the performance of classification problem.

In general, we divide the methods of classification problem in four groups as the following descriptions. 1) Statistical method: It was used in early classifier such as linear discriminate, quadratic discriminate, nearest neighbor, Bayes independence and Bayes second order. The Bayes' classifier was well known that has the least error classification rate. It is not practical in solving real world classification problem, since we need to know the probability density function of data previously. 2) Neural network: It is a system that is deliberately constructed to make use of some organizational principles resembling those of human brain such as [1-3] have good tasks. 3) Fuzzy inference engine: It mentioned the relation between classification problem and fuzzy set by Zadeh in [4]. Expert system identifies different pattern by the knowledge fuzzy rule database, which is set up by querying human expert experience or other techniques directly from training data. 4) Hybrid neural –fuzzy technique: It is one of the more promising approaches to computer-based pattern recognition [5-8].

Since we are not easy to find the experts, and who usually hard to express their knowledge. So, many different approaches extract knowledge directly from training data. These methods are based on neural networks or fuzzy set theory [9-20]. S. Abe and M-S. Lan extract the fuzzy rules from numerical data by recursively resolving overlaps between two classes [18-19]. Then, they said the optimal input variables for rules are determined using the number of extracted rules as criterion. But, there are still some drawbacks on this method such as following points. 1) It needs more computation time to recursively resolving overlaps between two classes. 2) It sometimes can't be resolved in some critical condition. 3) It can't update the rule structure on line.

Hong and Lee have pointed out that the drawbacks of most fuzzy controllers and fuzzy expert systems are that they need to predefine membership functions and fuzzy rules to map numerical data into linguistic terms and to make fuzzy reasoning work [20]. They proposed a method based on the fuzzy clustering technique and the decision tables to derive membership functions and fuzzy rules from numerical data. However, they still need to predefine the input variable smallest unit and it will take more computation time for constructing decision tables and merging operations as the attribute number and data scale becomes large.

Tzu-ping Wu and Shiyi-Ming Chen have a learning algorithm [21] based on the a -cuts of equivalence relations and a -cuts of fuzzy sets to construct the membership functions of the input variables and the output variables of fuzzy rules and to induce the fuzzy rules from numerical training data set. By experiment on Iris data, it shows the algorithm has a higher average classification ratio and can generate fewer rules than the existing algorithm. By this algorithm, we should predefine the a value to decide how many output linguistic labels will be generated. Then, there are still many a values must be selected as the number of input attribute is large or too many output linguistic labels were generated. That means we should decide many a values to create the input-value subsets for each input linguistic label of each linguistic variable. But, it didn't tell us how to select the a value.

Artificial neural networks have training and learning ability on line. And they have been successfully used in many pattern recognition problems [1-3]. But, this approach always likes a black box that can't be analyzed and explained in physical meaning. It usually lacks an ability to model the uncertain or ambiguous information existing among data, which is, so often encountered in the real world.

In this paper, we construct a high efficiency classifier by using the combination of fuzzy inference and neural network technology. In section II, we describe the definition of measure of fuzziness that will be used to restrict the neural network node making sense. The variable structure of fuzzy neural network will be showed in section III. The activation hyper-box, sub-inhibition hyper-box, total-inhibition hyper-box and loop feedback network node are defined. Then, we discuss the learning algorithm to get all parameters in this fuzzy-neural network in section IV. Finally, we show the performance for this high efficiency classifier to compare with other method in section V. We also make some conclusions in section VI.

II. Measure of Fuzziness

A. Definition of measure of fuzziness [22]

Two categories of uncertainty on data information can be recognized: vagueness and ambiguity. In general, vagueness is the uncertainty associated with difficulty of making a sharp or precise boundary in grouping objects of interest, while ambiguity is the uncertainty associated with choice, that is, difficulty in making a choice between two or more alternatives. Clearly, the concept of fuzzy sets provides a basic mathematical framework for dealing with vagueness. On the other hand, the concept of fuzzy measures provides a general mathematical framework for dealing with ambiguity. Hence fuzzy sets and fuzzy measures are tools for representing these two distinct forms of uncertainty. Measures of uncertainty related to vagueness are referred to measures of fuzziness.

In general, a measure of fuzziness is a function

$$f : \tilde{P}(x) \to R$$

where $\tilde{P}(x)$ denotes the set of all fuzzy subsets of X, R is the real line, and the function f satisfies the following axioms:

Axiom 1: $f(A) = 0$ if only if A is a crisp set.

Axiom 2: $A \prec B$, $f(A) \le f(B)$. Where $A \prec B$ denotes that A is shaper than B.

Axiom 3: $f(A)$ assumes the maximum values if only if A is maximally fuzzy.

Axiom 1 indicates that a crisp set has zero degree of fuzziness. Since there are different definitions of "shaper" in Axiom 2 and "maximally fuzzy" in Axiom 3, several different measures of fuzziness exist in the literature. In this paper, we are based on the following concept of "shaper" and "maximally fuzzy":

1. $a \prec B$ (A is shaper than B) is defined by

$$\begin{cases} m_A(x) \le m_B(x), & for \quad m_B(x) \le \dfrac{1}{2} \\ m_A(x) \ge m_B(x), & for \quad m_B(x) \ge \dfrac{1}{2} \end{cases} \qquad for \quad all \quad x \in X$$

2. A is maximally fuzzy if $m_A(x) = \dfrac{1}{2}$ for all $x \in X$

In this paper, the fuzzy measure that we want to introduce is defined by this function:

$$f(A) = -\sum_{x \in A} \{ m_A(x) \log_2 [m_A(x)] + [1 - m_A(x)] \log_2 [1 - m_A(x)] \} \qquad \text{……………………. …..(1)}$$

$$\hat{f}(A) = \frac{f(A)}{|X|} \qquad \text{………………… ……..(2)}$$

where $|X|$ denotes the cardinality of the universal set X. This measure of fuzziness can be considered as the entropy of a fuzzy set.

B. Measure of system fuzziness

Without loss of generality, we consider multi-input-single-output fuzzy logic system, since a multi-output system can always be decomposed into a group of single-output system. In this section, we define a classification system by a sequence of multi-input-single-output fuzzy rules. As following:

R_j : If x_1 is A_{1j} and x_2 is A_{2j} ……..and x_n is A_{nj} then the class is c_j

Where

n is the number of attribute of the classification system, i.e., the dimension of system.

c is the number of class of the system.

A_{ij} is the linguistic label.

$i = 1,2,....n$

$j = 1,2,....c$

By the T-norm operator with min operation, we can rewrite the system by the other symbols as following description:

$$R_j = \int_{(x_1, x_2, ... x_n)} m_{A_{1j}} \Lambda m_{A_{2j}} \Lambda ... m_{A_{nj}} /(x_1, x_2, ... x_n)$$

Then, according to the definition (1) we have

$$f(A_{ij}) = -\sum_{x_1 \in A_{ij}} \left\{ m_{A_{ij}}(x_1) \log_2 \left[m_{A_{ij}}(x_1) \right] + \left[1 - m_{A_{ij}}(x_1) \right] \log_2 \left[1 - m_{A_{ij}}(x_1) \right] \right\}$$

$$\hat{f}(A_{ij}) = \frac{f(A_{ij})}{|X_i|}$$

We also define the measure of fuzziness for rule R_j,

$$\hat{f}(R_j) = \sum_{i=1}^{n} \hat{f}(A_{ij}) \qquad (3)$$

and the total measure of fuzziness for the system

$$\hat{f}(R) = \sum_{j=1}^{c} \hat{f}(R_j) = \sum_{j=1}^{c} \sum_{i=1}^{n} \hat{f}(A_{ij}) \qquad (4)$$

This equation shows how fuzziness for a system description. Does any more rule description will let the system more clear? Some rules are good sufficient condition and don't need more redundancy rule to confuse the system. In next section, we will base on this concept to constructan optimal neural network structure for classifier.

III. Variable Neural-Fuzzy Network Structure

A. Min-Max points, Hyperbox Fuzzy Sets, and The membership Function

In this paper, we use the hyper-box definition in [7]. We normalize the range of each dimension from 0 to 1; hence the pattern space will be the n-dimensional unit cube I^n. The membership function for each hyper-box fuzzy set must describe the degree to which a pattern fits within the hyper-box. In addition, it is typical to have the membership values range between 0 and 1. Let each hyper-box fuzzy set; B_j be defined by the order set

$$B_j = \{X, V_j, W_j, f(X, V_j, W_j)\} \qquad \forall X \in I^n \qquad (5)$$

Using this definition of a hyper-box fuzzy set, the aggregate fuzzy set that defines the k th pattern class C_k is defined as

$$C_k = \bigcup_{j \in K} B_j \qquad (6)$$

where K is the index set of those hyper-boxes associated with class k. Note the union operation in fuzzy set is typically the max of all of the associated fuzzy membership functions.

The membership function $b_j(A_h)$ for the j th hyper-box, $0 \le b_j(A_h) \le 1$, must measure the degree to which the h th input pattern A_h falls from the center of the hyper-box B_j. On a dimension-by-dimension basis, this can be considered a measurement of how far each component from the center of the hyper-box. Also, as $b_j(A_h)$ approaches 1, the point should be more near by the center of the hyper-box B_j. With the value 1, it represent the point is the center of the hyper-box exactly. And, the edge of the hyper-box should be assigned the value near 0. The function that meets all above criteria is defined as following equations:

$$b_j(A_h) = \frac{1}{n}\sum_{i=1}^{n}\left\{\min(\min(1, \max(0, \frac{a_{hi} - v_{ji} - d(w_{ji} - v_{ji})}{c_{ji} - v_{ji} - d(w_{ji} - v_{ji})})), \min(1, \max(0, \frac{w_{ji} + d(w_{ji} - v_{ji}) - a_{hi}}{w_{ji} + d(w_{ji} - v_{ji}) - c_{ji}})))\right\} \quad ..(7)$$

where $A_h = (a_{h1}, a_{h2}, \ldots a_{hn}) \in I^n$ is the h th input pattern, $V_j = (v_{j1}, v_{j2}, \ldots v_{jn})$ is minimum point for B_j, $W_j = (w_{j1}, w_{j2}, \ldots w_{jn})$ is maximum point for B_j, and d is the sensitivity parameter that regulates how much sample data in the hyper-box. The more data support in this hyper-box, the value of d more closed to 0. But, it is not really critical for the classification rate. It will be selected by any value below 0.1. The connections are adjusted using the algorithm described in section IV.

B. Implementing The variable Neural -Fuzzy Network Structure

Let us review firstly on general min-max neural network in [7] that shows the structure on fig 1. This structure has 3 layers: the first layer supports n attributes for testing data input, the second layer is configured by m hyper-boxes, the third layer has c nodes for different classes. Using the definition of hyper-box in [7] or [8], we find that each hyper-box should be cut very clearly. And, this unnecessary cut result in error sometimes. Anyway, this structure always uses too many hyper-box nodes in second layer that means too much cost to implement this system. We will conquer this drawback on next paragraph.

We propose another variable structure in this paper that shows on Fig 2. The first and third layer in fig 2, it same as in fig 1. There are only c hyper-box nodes in second layer and increase one feedback node f^l in third layer. This change will save cost on hardware implementation, because we don't need too many second layer node. We will save the computation time on software implementation, since it just need less computation unit than Fig 1. Each hyper-box node can be represented by function $b_j^l(X) = F(X, V, W, l)$, called l th loop hyper-box function or l th level hyper-box function. The feedback node f^l feedback the value $b_j^l(X)$ for next loop until the feedback criteria is satisfied. We will discuss this criteria based on measure of fuzziness later.

C. l th loop hyper-box function

For saving the second layer unit cost, we divide the universe space in the least hyper-boxes. If we got c classes sample data, we have c hyper-box in the zero loops (that means initial value b_j^0 will be assigned firstly). To solve overlaps between different classes, we introduce two types of hyper-boxes, which were discussed in [18]: activation hyper-boxes, which define the existence regions for classes, and inhibition hyper-boxes, which inhibit the existence of data within the activation hyper-boxes. These hyper-boxes are defined recursively. First we determine activation hyper-boxes by calculating the minimum and maximum values of data for each class. If the activation hyper-box for class i overlaps with any other activation hyper-boxes for class j, the overlapping region is defined as a sub-inhibition hyper-box. We also define the union of all sub-inhibition hyper-box as a main- inhibition hyper-box. If the main- inhibition hyper-box is existed, then we will find next loop activation hyper-boxes b_j^1. The more loops generate, the more small activation hyper-boxes will be found. If we didn't give any limitation, we may find an activation hyper-box include only one sample data. But it is no meaning to construct so small hyper-box, because that sample data may be just a noise or some uncertainty. We don't need this uncertainty or ambiguous data to become so clear information in our structure and to waste our computation time. So, we should define some criteria to stop the recursively finding loop hyper-box function procedure.

We assume the l th loop main- inhibition hyper-box can be divided to c activation hyper-boxes, then we can set a c-rules fuzzy system denoted by R^l. Now, we check the fuzziness of this system by equation (4) as

$$\hat{f}(R^l) = \sum_{j=1}^{c}\hat{f}(R_j^l) = \sum_{j=1}^{c}\sum_{i=1}^{n}\hat{f}(A_{ij}^l) \quad \ldots\ldots(8)$$

We also set a a -cut value for the limitation of fuzziness measure. If $\hat{f}(R^l) > a$ is true, then we don't need the l th loop hyper-box function b_j^l any more. It means that we only have l different function parameters in second layer of this fuzzy-neural network.

D. The feedback node function f^l and c class nodes function c_k in third layer

The feedback node in third layer is designed to change the second layer parameter. The connections between the third layer and second layer are binary valued and stored in the matrix U. The equation for assigning the values to the connections is

$$u_{jk} = \begin{cases} 1 & if \quad b_j \quad is \quad a \quad hyper-box \quad for \quad class \quad c_k \\ 0 & otherwise \end{cases} \qquad \cdots\cdots(9)$$

where b_j is the j th node in second layer and c_k is the k th node in third layer. Each node in third layer represents a class. The output of node represents the degree to which the input pattern A_h fits within the class k. The transfer function of output node is defined as

$$c_k = \max_{j=1}^{c} b_j u_{jk} \qquad \cdots\cdots(10)$$

There are two main ways that the output class nodes can be utilized. If a soft decision is desired, the output is utilized directly. If a hard decision is required, the output node with the highest value is located.

This final result is depend on the second layer node value. We can find the two highest values of output node values c_k are c_{k1} and c_{k2}. If the difference of these two values is smaller than the value b, the second layer parameters should be changed until we don't have next loop values. So, if $|c_{k1} - c_{k2}| > b$ is true, then we can stop to change next loop parameter.

IV. The Algorithm to find hyper-boxes

In this section, we describe the recursive definition of activation and inhibition hyper-boxes by 2 dimensions, 2 classes example that shows on fig 3. We define the intersection area that nominate inhibition hyper-box for loop1 denoted by I(1). We will find the activation hyper-boxes in loop1. an activation A_{ii}^l, is defined, which is the maximum region of class i data $A_{ii}^l = \{x | V_{ii}^l \le x_k \le W_{ii}^l, \quad k = 1,....,m, \quad x \in X_i\}$

Where

x_k : the k th element of x;

V_{ii}^l : the minimum value of x_k for $x \in X_i$;

W_{ii}^l : the maximum value of x_k for $x \in X_i$

We also can find another class activation hyper-box denoted by A_{jj}^l. If there is no overlap between activation hyper-boxes A_{ii}^l and A_{jj}^l, we don't need to find next loop activation hyper-box. If the activation hyper-boxes A_{ii}^l and A_{jj}^l overlap, we resolve the overlap recursively as illustrated in fig 3 in which we define the overlapping region as sub-inhibition box of loop l denoted as $I(l)$.

For n dimensional data and c classes problem, we will follow above procedure to set all activation hyper-boxes and sub-hyper-boxes. We will connect all sub-hyper-boxes for each loop to become a main-hyper-box. We resolve the main-hyper-box to become many activation hyper-boxes. It is a recursive procedure until the condition $\hat{f}(R^l) > a$ be satisfied in equation (8).

V. Performance Evaluation

We will show performance of this classifier by different wine classes on real word data. We also show the result that compare with other methods. All datum come from University of California, Irvine database.

There are three classes of wine for 178 record data. Each record data has 13 attributes (Alcohol、Malic acid、Ash、Alcalinity of ash、Magnesium、Total phenols、Flavanoids、Nonflavanoid phenols、Proanthocyanins、Color intensity、Hue、OD280/OD315 of diluted wines、Proline) which show on fig 4(59 records for first class, 71 records for second class, 48 records for third class).

First, we set up the fuzzy-neural network by training all 178 data records and test by the same data. Using same data, Corcoran 和 Sen [23] generate 60 non-fuzzy if-then rules via other learning system based on genetic algorithm. The result shows as following: optimum classification rate 100%, average classification rate 99.5%, the worst classification rate 98.3%.

Ishibuchi, Nakashima and Murata[22] proposed lattice-separation method also using the same data. They set up the fuzzy inference engine via 5 linguistic terms and one "don't care" term to be fuzzy rule premise. Since there are 13 attributes for the wine data, it may generate 6^{13} rules for this training algorithm. The best 60 rules will find by the genetic algorithm. The result shows as following: optimum classification rate 99.4%, average classification rate 98.5%, the worst classification rate 97.8%.

In this paper, we test all data by our method. Since there are three classes of wine, we got three activation hyper-boxes in 0 loop. The result shows as following: optimum classification rate 100%, average classification rate 100%, the worst classification rate 100%.

VI. Conclusion

In this paper, we propose a variable fuzzy-neural structure network that constructs by 3 layers for pattern classification. The first layer is for data input, and third layer is for output decision. We make the second layer of network, which is set by each activation fuzzy hypercube for each class. The fuzzy hypercube is an n-dimensional box defined by a min point and max point with a corresponding membership function. Then, we test the condition of overlap of these hypercube by defined tuning-structure parameter, which is made by fuzzy measure, information support degree and data separation degree. We decide whether the second layer of the network should be rebuild again. Repeat above procedure, we can generate a high efficiency classifier by this dynamic neural-fuzzy network structure using sufficient information of all training data. We also decrease the complexity of classification computation according to different test data.

Reference

1. J. C. Bezdek, (Sept. 1992), "Computing with uncertainty", IEEE Commun. Mag., pp. 24-36.
2. M. H. Hassoun, (1995), Fundamentals of Artificial Neural Networks. Cambridge ,MA:MIT Press.
3. P. K. Simpson, (1990), Artificial Neural systems: Foundations Paradigms, Applications, and Implementations. New York:Pergamon.
4. L. A. Zadeh, (1965), Fuzzy sets, Inform. And Control 8 , pp. 338-353.
5. F. Uebele, S .Abe, and M-S. Lan, (1995), "A Neural Network-Based Fuzzy Classifier," *IEEE Tran. Systems,Man,and Cybernetics*, vol. 25, no. 2, pp. 353-361.
6. S. Abe, (1996), *Neural Networks and Fuzzy Systems: Theory and Applications*, Kluwer Academic, Boston.
7. P. K. Simpson, (Sep. 1992), "Fuzzy min-max neural networks—Part 1：Classification," *IEEE Trans. Neural Network*, vol. 3, pp. 776-786.
8. B. Gabrys and A. Bargiela, (May. 2000), "General fuzzy min-max neural network for clustering and classification," *IEEE Trans. Neural Network*, vol. 11, pp. 769-783.
9. A. Grauel and H. Mackenberg, (1997), "Mathmatical analysis of the Sugeno controller leading to general design rules," Fuzzy Sets Syst. , Vol. 85, no. 2, pp. 165-175.
10. F. Klawobnn and R. Kruse, (1997), "Constructing a fuzzy controller from data," Fuzzy Sets Syst. , Vol. 85, no. 2, pp. 177-193.
11. K. Nozaki, H. Ishibuchi, and H. Tanaka, (1997), "A Simple but powerful heuristic method for generating fuzzy rule from numerical data," Fuzzy Sets Syst. , Vol. 86, no. 3, pp. 251-270.
12. L. -X. Wang and J. M. Mendel, (Nov./Dec. 1992), "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern*., vol. 22, pp. 1414–1427.
13. C. T. Lin and C. S. George Lee, (1991), Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Comput*, vol. 40,pp. 1320-1336.

14. H. Ishibuchi and H. Tanaka, (1993), "Neural network that learn from fuzzy if-then rules," *IEEE Trans. Fuzzy Systems*, vol. 1,pp. 85-97.
15. J. J. Shann and H. C. Fu, (1995), "A fuzzy neural network for rule acquiring on fuzzy control systems," Fuzzy Sets Syst., Vol. 71, no. 3, pp. 345-357.
16. R. Langari and L. Wang, (1996), "Fuzzy Models, modular networks, and hybrid learning," Fuzzy Sets Syst., Vol. 79, no. 2, pp. 141-150.
17. N. K. Kasabov, (1996), "Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid system," Fuzzy Sets Syst., Vol. 82, no. 2, pp. 135-149.
18. S. Abe and M-S. Lan, (Feb 1995), "A Method for Fuzzy Rules Extraction Directly from Numerical Data and Its Application to Pattern Classification," *IEEE Trans. Fuzzy Systems*, vol. 3, No. 1, pp. 18-28.
19. S. Abe and M-S. Lan, (Jan. 1995), "Fuzzy rules extraction directly from numerical data for function approximation," *IEEE Trans. Syst., Man, Cybern*., vol. 25, pp. 119–129.
20. T.P. Hong and C.Y. Lee, (1996), "Induction of Fuzzy rules and membership functions from training examples," Fuzzy Sets Syst., Vol. 84, no. 1, pp. 33-47.
21. Tzu-ping Wu and Shiyi-Ming Chen, (Feb. 1999), "A new method for constructing membership functions and fuzzy rules from training examples," *IEEE Trans.on Systems , Man, and Cybernetics—Part B: Cybernetics*, Vol. 29, No 1.
22. C.T. Lin and C.S. George Lee, Neural Fuzzy Systems, Prentice Hall, 1996.
23. J. C. Bezdek, (1973), *Fuzzy Mathematics in Pattern Classification*. Phd thesis, Applied Math. Center, Cornell University, Ithaca.
24. S. E. Papadakis and J. B. Theocharis, (2002), "A GA-based fuzzy modeling approach for generating TSK models," *FUZZY Set And Systems*, pp. 1-32.
25. S. Abe, (Apr. 1998), "Training of Fuzzy Classifier with Ellipsoidal Regions by Dynamic Cluster Generation," *Second International Conference on Knowledge-Based Intelligent Electronic Systems*, pp. 21-23.
26. I. H. Suh, J. –H. Kim and F. C. –H. Rhee, (jun. 1999), "Convex-set-based fuzzy clustering," *IEEE Trans. On Fuzzy Systems*, vol. 7, no. 3, pp. 271-28.
27. P. J. Werbos, (1974), "Beyond regression：New tools for prediction and analysis in the behavioral sciences," *Doctoral dissertation, Harvard University*.
28. H. M. Lee, C. M. Chen, J. M. Chen, and Y. L. Jou, (Jun. 2001), "An efficient fuzzy classifier with feature selection based on fuzzy entropy," *IEEE Trans. Syst., Man, Cybern*., Vol. 31, No. 3.

91 年 09 月 12 日投稿
91 年 10 月 02 日接受

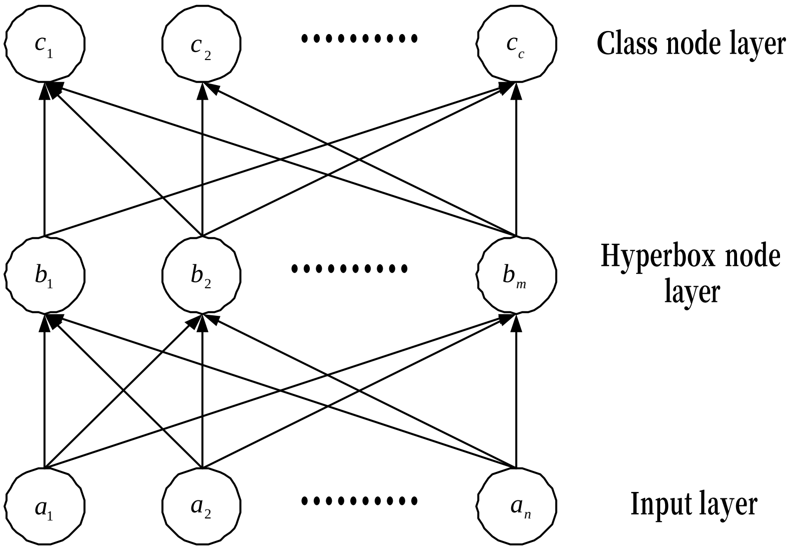

Fig 1

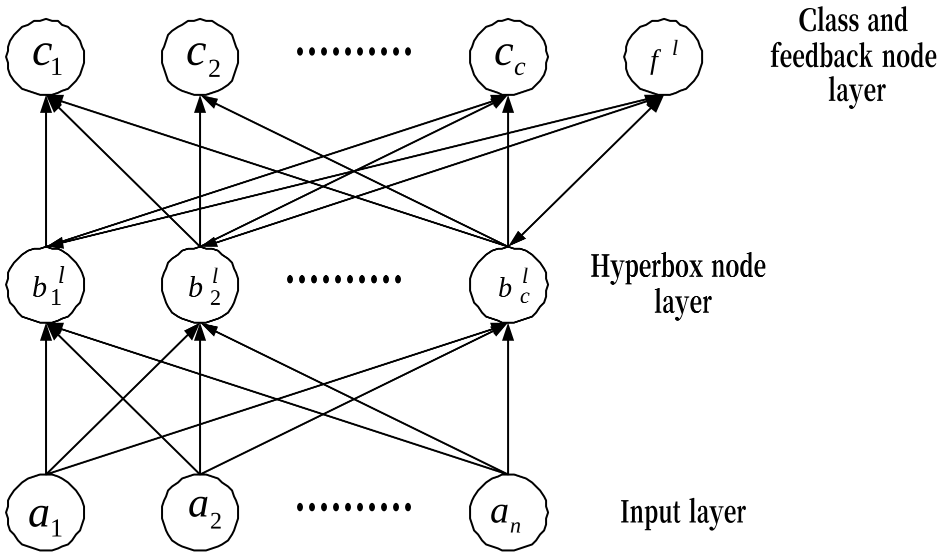

Fig 2

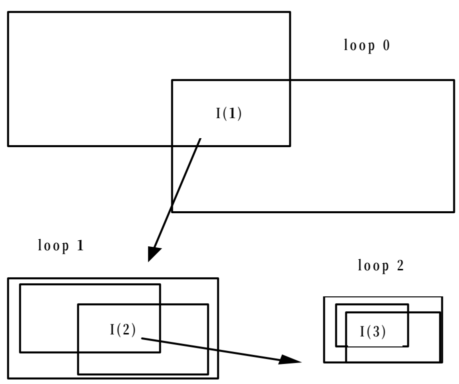

Fig 3

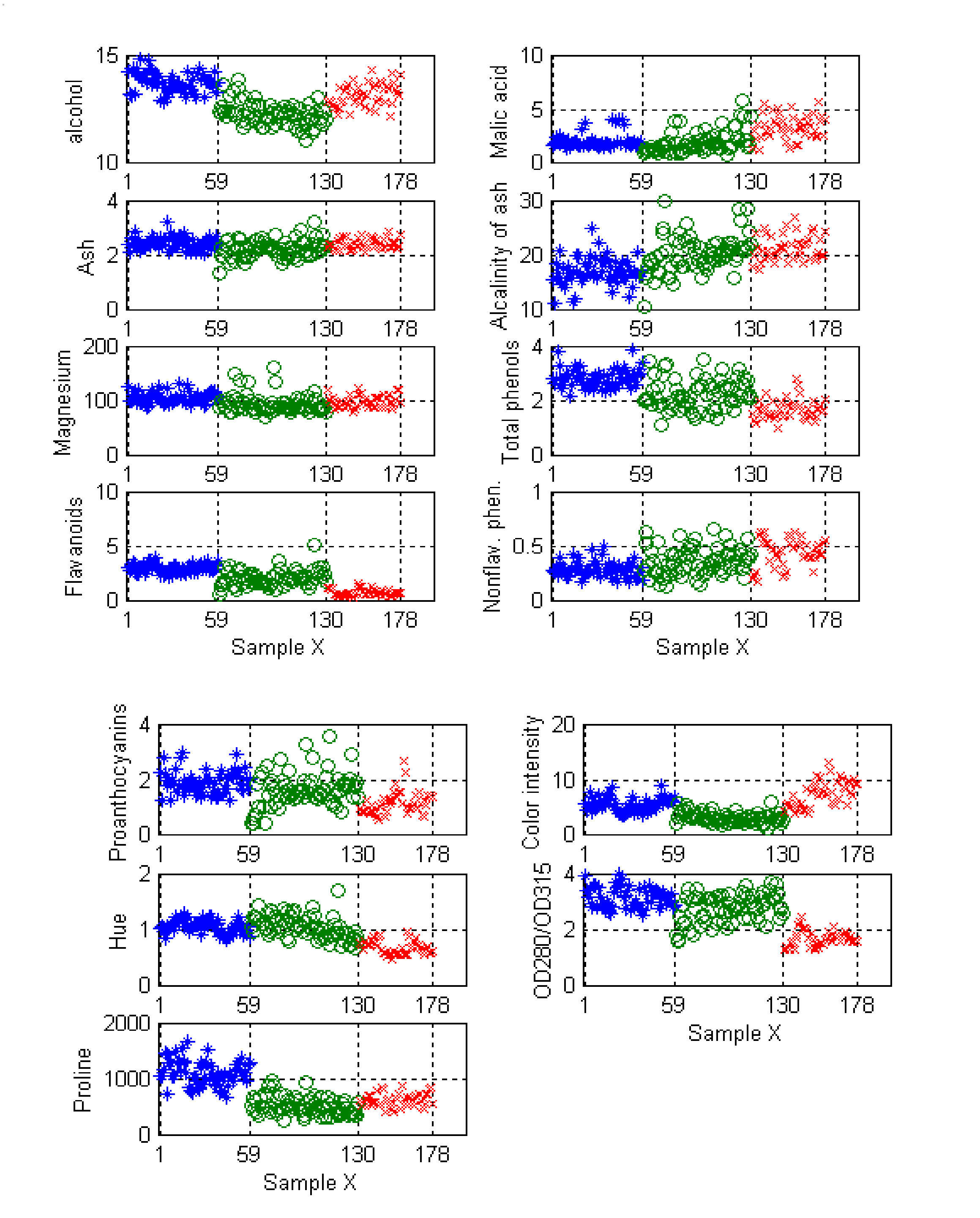

Fig 4　13 attributes data distribution for 3 classes of wine.