Enhance Neuro-Fuzzy System for Classification Using Dynamic Clustering

Poonarin Wongchomphu
Computer Engineering Department
Faculty of Engineering, Chiang Mai University
Chiang Mai, Thailand
poonarin@gmail.com

Narissara Eiamkanitchat
Computer Engineering Department
Faculty of Engineering, Chiang Mai University
Chiang Mai, Thailand
narisara@eng.cmu.ac.th

Abstract—**The Enhance Neuro-fuzzy system for classification using dynamic clustering presents in this paper is an extension of the original Neuro-fuzzy method for linguistic feature selection and rule-based classification. The new algorithm resolves the limitations of the original algorithm that uses only 3 membership functions for all features to fine the appropriate function for each feature. Each feature of the dataset is pre-processed by a new approach to clustering automatically. The Neuro-fuzzy classification models for each dataset is created in accordance with the number of clusters have been divided for each feature. In order to be appropriate functioning in the Neuro-fuzzy structure, a new algorithm has been adapted to use the binary instead of the bipolar as original algorithm. Thirteen datasets were used to test the performance of the proposed algorithm. The average accuracy calculated from the 10-fold cross validation found that this method can increase performance of the already proof high accuracy Neuro-fuzzy for classification.**

Keywords-component; Neuro-fuzzy; Classification; Dynamic Clustering

I. INTRODUCTION

The development of classification algorithms that can be predicted with high accuracy has been attracting researchers continuously. The reason is these techniques have been applied widely in various areas. There are many techniques for the classification. One of an interesting algorithm is Neuro-fuzzy which integrated two outstanding algorithms, neural network and fuzzy logic together.

The fuzzy logic algorithm is an appropriate method when applied to the data analysis which segmentation looks uncertain. Framework of fuzzy logic supports that analysis procedures can easily be interpreted by a rule extraction. The learning process does not support tuning parameters to identify the significance of the data [1]. The neural network algorithm is robust approach to the dataset that has never learned before. However, the original model of this method is difficult to interpret. In addition, the models are sensitive to variations in the dataset to alter its performance depends on the input data pre-processing [2-3].

Currently Neuro-Fuzzy technique was applied in the analysis of data and widely used in various areas. For example, in [1-2] have introduced techniques for data classifications to assist in the interpretation of data to increase productivity in the industry. The Neuro-fuzzy system is used to forecast wind speed and direction in [6] to analyze data for decision support in medicine [7] and is also used for control and decision support part in robotic [8 -10].

Techniques for data classifications using Neuro-Fuzzy has been continually evolving to ensure efficient classification accuracy. In [11-12] have proposed Neuro-fuzzy method for linguistic feature selection and rule-based classification by applying fuzzy logic techniques together with the neural network. This model outperforms other classification algorithms can also be extracted as rules to be used for classifications as well. The framework defines fixed three membership functions for converting the original input to linguistic variables. The new variables are forwarded to the neural network to learn and adjust the appropriate parameters for the further classification with high accuracy. The classification results when applied to a variety of datasets showed a high classification performance. However using this method on datasets with more than three classes the classification performance is not satisfactory.

This research focuses on the design and development of Neuro-fuzzy framework presented in [12]. The new concept proposes in order to determine the number of membership functions and initial values of the parameters needed in functions for each considered feature automatically. In addition, the proposed method also improves the learning process in order to enhance the Neuro-Fuzzy to provide a classification of datasets with multiple classes more accurate.

II. PROPOSED WORK

The new method presented here can divide the process into three main parts. Firstly, the dynamic clustering process is performed to find the number of membership functions appropriate to each feature. The second part is the transition process to transform the original input data through a Gaussian membership function to generate a binary input. Lastly, the classification process is part of the learning and classification using neural network.

978-1-4799-3855-1/14/$31.00 ©2014 IEEE

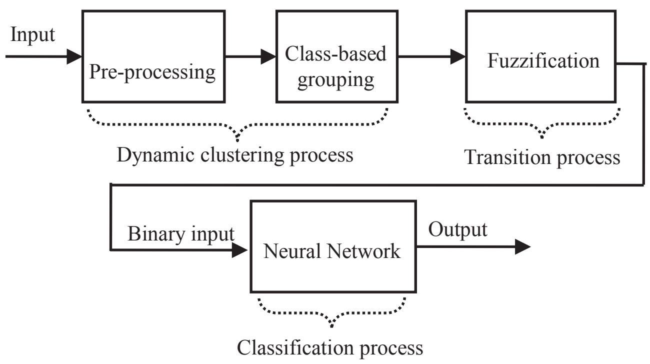

Figure 1. The structure of the enhance Neuro-fuzzy system for classifications using dynamic clustering

The mechanism of the total structure shown in Figure 1 is described in more detail in subsequent sections.

A. Dynamic Clustering Process

The first step in the process of data analysis in general is to create the scatter plot. The distribution of data in each class can be observed in order to determine the categorical number of each feature.

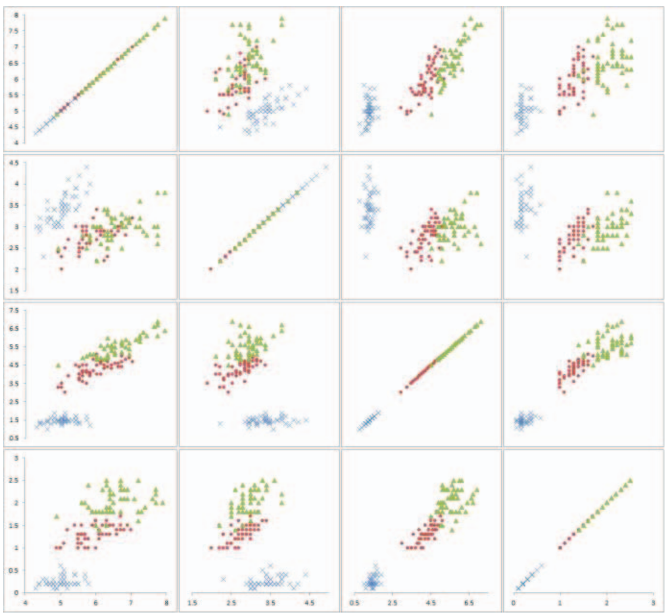

Figure 2. The features scatter plot of iris dataset

Noticeable in Figure 2 that the distribution of data in feature 3 and feature 4 can be categorized into 3 groups according to class in the dataset properly. Graphs in row 1 and row 2 belong to feature 1 and feature 2, in some graphs the relationships cannot be separated easily.

Determining a fixed number of categories and applies to all features not appropriate in some datasets. Obviously the good criteria to specify the number of categories should consider the relationship of the feature distribution and the class of the data set. According to the class in the dataset and the number of cluster, this paper introduces a new approach to perform dynamic clustering by referencing to a class of the dataset. The main idea is to group the data in the same class with the smallest distance value. In order to deal with the noise, the method of calculating the threshold is proposed for exceeding cluster eliminating. The ideas described above resulted in a number of clusters that is appropriate for the classification precision.

Algorithm : Class-Based Grouping Algorithm

Input :
x_i : Input feature i.
Output :
N_i : Number of membership function of input feature i.
c_i : mean of each membership function of input feature i.
σ_i : standard deviation of each membership function of input feature i.

Begin
x_i = Sort(x_i);
j = 1
N = 1;
prevClass = Class of x_{i1};
x_{ij} is a member of group 1;
For j = 2 to Number of record
  If *prevC* != C_j Then
    $N = N + 1$;
    prevClass = Class of x_{ij};
  End If
  x_{ij} is a member of group n;
End For

threshold = Average of number of member of each group;
Calculate centroid of each group;
repeat
  For j = 1 to N
    If Number of group j < *threshold* Then
      Merge group j to other group that has nearest centroid and calculate new centroid;
      $N = N - 1$;
    End If
  End For
until No group that has a number of member less than *threshold*
Calculate c of each group;
Calculate σ of each group;
Return(N, c_i, σ_i);
End

Figure 3. Class-Based Grouping Algorithm

The algorithm for dynamic clustering is called "Class-based Grouping" as displayed in Figure 3. The example of the algorithm implements with the feature i can be explained as following section.

step :1

x_i	Class	
4.40	1	Cluster 1
4.75	1	
4.83	1	
4.98	2	Cluster 2
5.01	1	Cluster 3
5.06	2	Cluster 4
5.13	2	
5.18	2	
5.24	3	Cluster 5
5.29	2	Cluster 6
5.32	2	
5.41	3	Cluster 7
5.43	3	
5.53	3	
5.60	3	

step :2

Group	Centroid	*n* member
1	4.66	3
2	4.98	1
3	5.01	1
4	5.12	3
5	5.24	1
6	5.31	2
7	5.49	4
threshold		2.125

step :3

Group	Centroid	*n* member
1	4.66	3
2,3,4	5.07	5
5,6	5.28	3
7	5.49	3

step :4

Cluster	*c*	*σ*
1	4.66	0.186
2,3,4	5.07	0.074
5,6	5.28	0.032
7	5.49	0.076

Figure 4. Example of dynamic clustering

The example of class-based grouping is demonstrated in Figure 4. There are 4 working steps in this algorithm.

The first step begins with sorting data point in feature x_i in ascending order. The process is similar to the incremental clustering the first data point is assigned to cluster 1. Then define the class of data in the first cluster as the original class of the data. The class of each sample is compared to the class of previous cluster. In case that the class is the same as previous one then assigned sample to previous cluster, otherwise create a new cluster. All processes are iterated until all samples in dataset are considered and assign cluster.

In the second step all centroids of each cluster are calculated. In order to apply the agglomerative clustering each feature must have proper threshold, which can calculate from the equation (1) as following.

$$F_{th} = \frac{\sum_{j=1}^{N} M_j}{N}. \qquad (1)$$

The F_{th} is the threshold of considered feature which is the average number of members in each cluster. The variable M_j is the number of members in cluster j. The N represents the total numbers of clusters derive from the first step. The F_{th} is used to judged existence of the cluster in the third step.

The third step, all clusters which have member in cluster less than F_{th} are eliminated. All eliminated clusters are reconsidered to combine with other clusters based on the less distance between centroids of clusters. As displayed on the right side of Figure 4, in step 3, the member in group 2 is less than the threshold then considered agglomerative clustering. Comparison the centroid distance, the distance between centroids of group 2 and centroid of group 3 is less than the distance between centroids of group 2 and centroid of group 1. Group 2 is considered combine with group 3. The distance function of centroid can calculate from the equation (2) as

$$d(c_j, c_k) = \left| c_j - c_k \right| \qquad (2).$$

The c_j in the equation (2) represent the centroid of the eliminated cluster. The c_k is the centroid of other clusters. The cluster to be combined with is judged from the k cluster that relevant to the results of J which is the minimum value of distance in the equation (3).

$$J = \min\{d(c_j, c_k)\}; k = 1, \ldots (N-1) \qquad (3).$$

The process in the third step is recursive until all clusters have member more than the threshold value.

The fourth step all clusters are prepared to created Gaussian membership function by calculate mean(c) and standard deviation (σ) of each cluster.

The dynamic clustering has 4 sub-processes as described previously. The algorithm is applied to each feature in the dataset. The final cluster number of each feature does not have to be the same number. The number of clusters can vary from 1 cluster to any number depending on the distribution of value in the feature. The feature that has distribution value correlated to class, number of clusters almost the same as a number of classes. The feature that has distribution value is not correlated to class, number of clusters usually higher than a number of classes.

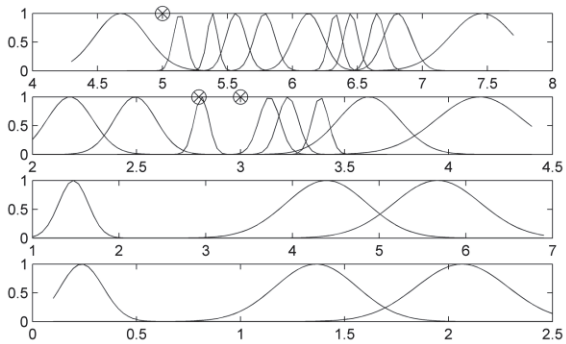

Figure 5. Membership function of feature 1 to 4 of Iris dataset from the proposed method

Figure 5 shows the Iris membership functions correspond to the result of the dynamic clustering algorithm. The resultant number of clusters in feature 3 and feature 4 are the same as a number of classes. The resultant number of clusters in feature 1 and feature 2 are 12 clusters which are higher than a number of classes. The Iris dataset is the well-known in data analysis fields that the feature 3 and feature 4 are important some algorithms can use only these features to perform the high accuracy classification results.

B. Transition Process

The number of clusters and initial values of the mean and standard deviation from a dynamic clustering process are used in the transition process. The original inputs are fed to the Gaussian membership function layer of the model. The number of membership functions for each feature is varies depending on the cluster of the feature.

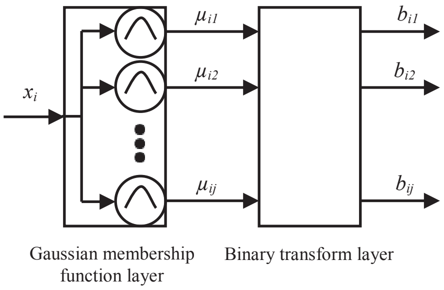

Figure 6. The structure of transition process by fuzzy membership function of feature *i*

The structure of transition processes is shown in Figure 6. The structure in Figure 6 including Gaussian membership function layer and Binary transform layer. The membership values of feature x_i are defined in equation (4).

$$\mu_{ij} = \begin{cases} 0, & \text{if } \sigma_{ij} = 0 \text{ and } x_i \neq c_{ij}, \\ e^{\left(-\frac{1}{2}\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right)}, & \text{if } \sigma_{ij} \neq 0, \\ 1, & \text{if } \sigma_{ij} = 0 \text{ and } x_i = c_{ij}, \end{cases} \tag{4}$$

where *i* is identified the original feature and *j* is identified the cluster order of each feature. Normally the membership value calculated from Gaussian membership function. However the membership value is set to 0 if no distribution of that cluster and x_i is not equal to mean of the cluster. The membership value is set to 1 if no distribution of that cluster and x_i is equal to mean of the cluster.

All results from the first layer are fed to the binary transform layer. The membership values are transformed into a binary value according to equation (5).

$$b_{ij} = \begin{cases} 1, & \text{if } \mu_{ij} \text{ is maximum}, \\ 0, & \text{otherwise}. \end{cases} \tag{5}$$

The maximum membership value means that the original feature is belongs to that cluster then set to 1. The others membership values are set to 0. The results from the binary transform layer are fed to the output layer for classification processes.

C. Classification Process

The structure of the classification process is shown in Figure 7. The binary inputs are fed to the neural network in the output layer.

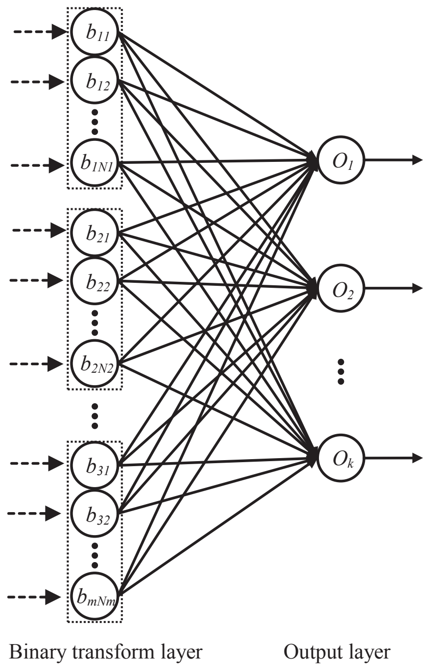

Figure 7. Structure of classification process.

The nodes in binary transform layer consider as input layer are equal to the number of clusters (N_m) from each feature (m). The nodes in the output layer are equal to the number of classes in the dataset. The sigmoid function is used as activation function in every node of the output layer. The output value can calculate from the following equations.

$$O_k = \frac{1}{(1 + e^{-S_k})}, \tag{6}$$

where $$S_k = \sum_{i=1}^{m} \sum_{j=1}^{N_i} w_{kij} b_{ij}. \tag{7}$$

In equation (7) *k* is the node in output layer, *j is* identified node of input layer and *i* identified the original feature. The w_{kij} is the weight of feature *i* that link between node *k* of output layer and node *j* of input layer.

The concept of back propagation to minimize the error is used in learning phase. The error signal of output node k at iteration p are defined by

$$e_k(p) = T_k(p) - O_k(p). \tag{8}$$

where e_k (p) is the error signal, T_k (p) is the target output, and O_k (p) is the output signal of node k. The delta value of output layer is define as

$$\delta^o(p) = e_k \varphi'(S(p)). \tag{9}$$

All weight in output layer are update by

$$w_{kji}(p+1) = w_{kij}(p) + \Delta w_{kij}(p), \tag{10}$$

where

$$\Delta w_{kij}(p) = -\eta \delta^o(p) b_{ij}(p). \tag{11}$$

III. EXPERIMENTAL RESULTS

The performance of the enhance Neuro-fuzzy system for classification using dynamic clustering present by applying an algorithm to implement with 13 various datasets from UCI [13]. The details of datasets are described in the table I.

TABLE I. DATASET USED IN EXPERIMENTS

Dataset	Size	Feature	Feature Type	Class
Breast Cancer	699	9	Integer	2
Forest Fire	517	12	Real	2
Glass Identification	214	9	Real	7
Hayes - Roth	160	4	Categorical	3
Ionosphere	315	34	Integer, Real	2
Iris	160	4	Real	3
Liver Disorders	345	6	Categorical, Integer, Real	2
Vote (Congressional Voting Records)	435	16	Categorical	2
Pima Indians Diabetes	768	8	Integer, Real	2
Seeds	210	7	Real	3
Wine	178	13	Integer, Real	3
Yeast	1484	8	Real	10
Zoo	101	16	Categorical, Integer	7

The performance of the proposed method is tested for accuracy compared with the algorithm Neuro-fuzzy [12]. The algorithm of Neuro-fuzzy in [12] is used as our benchmark because it has been proven that a highly accurate generalization among various algorithms, and is compatible with a wide variety of data types. In testing both algorithms using same parameters as possible, and the results shown in the table II are calculated from the average accuracy of the 10 fold cross-validation.

TABLE II. THE COMPARISON OF CLASSIFICATION RESULTS OF THE PROPOSED METHOD AND METHOD IN [12]

Dataset	Average Accuracy of 10-fold cross validation	
	Neuro-fuzzy from[12]	Proposed work
Breast Cancer	96.99%	**97.13%**
Forest Fire	**58.23%**	55.50%
Glass	56.67%	**67.25%**
Hayes	61.26%	**74.23%**
Ionosphere	**93.17%**	92.04%
Iris	96.00%	**96.67%**
Liver Disorders	**67.91%**	65.54%
Pima	85.16%	**86.85%**
Seeds	88.57%	88.57%
Vote	95.18%	95.18%
Wine	95.49%	**97.19%**
Yeast	19.20%	**51.08%**
Zoo	94.18%	**94.27%**
Average	77.54%	**81.65%**

It is evident from Table II, the accuracy of 8 datasets from the proposed algorithm the enhance Neuro-fuzzy system for classification using dynamic clustering are higher than Neuro-fuzzy model in [12]. In particular, the following set of Glass Hayes and Yeast, which has 7, 3 and 10 classes, the accuracy of the proposed algorithm is 10.58%, 12.97% and 31.88% higher, respectively.

IV. CONCLUSION

The new method of the enhance Neuro-fuzzy system for classifications using dynamic clustering shows the impressive high accuracy of classification than the Neuro-fuzzy method for linguistic feature selection and rule-based classification [12]. An algorithm in [12-13] yields higher accuracy than the many other methods of classification. A significant addition is the dynamic clustering that can determine the appropriate group for each feature in the dataset. Experimental results have demonstrated that the appropriate membership functions can result the higher correct classification and can be applied to applications in various areas of data types.

REFERENCES

[1] J. Han, M. Kamber and J. Pei, "Data Mining: Concepts and Techniques," Morgan Kauf-mann, San Francisco, 2006. pp. 325-370.

[2] S. S. Haykin, “Neural Networks and Learning Machines,” 3rd Ed., Prentice Hall, New York, 2009.

[3] D. Kriesel, “A Brief Introduction to Neural Networks,” Retrieved August, 15, 2011. pp. 37-124.

[4] W. Chia-Ching, C. Thao-Tsen and L. Shie-Jue, “A k-NN Based Neuro-Fuzzy System for Time Series Prediction,” In: Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2013 14th ACIS International Conference on. IEEE,2013. pp. 569-574.

[5] C. Affonso and R. J. Sassi, “A Rough-Neuro Fuzzy Network applied to polymer processing,” In: Software Engineering and Data Mining (SEDM), 2010 2nd International Conference on. IEEE, 2010. pp. 355-360.

[6] K. Subramanian, R. Savitha and S. Suresh, “Complex-valued neuro-fuzzy inference system for wind prediction,” In: Neural Networks (IJCNN), The 2012 International Joint Conference on. IEEE, 2012. pp. 1-7.

[7] T. Kondo, J. Ueno and S. Takao, “Hybrid feedback GMDH-type neural network self-selecting various neurons and its application to medical image diagnosis of lung cancer,” In: Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on. IEEE, 2012. pp. 1925-1930.

[8] A. Gajate, R. E. Haber, P. I. Vega and J. R. “Alique A transductive neuro-fuzzy controller: application to a drilling process,” Neural Networks, IEEE Transactions , vol. 21, no. 7, pp. 1158-1167, July 2010.

[9] E. Kayacan, H. Ramon, W. Saeys, “Adaptive Neuro-Fuzzy Control of a Spherical Rolling Robot Using Sliding-Mode-Control-Theory-Based Online Learning Algorithm,” Cybernetics, IEEE Transactions , vol. 43, no. 1, pp. 170-179, Feb. 2013.

[10] H. Rouabah, C. Abdelmoula and M. Masmoudi, “Behavior control of a mobile robot based on Fuzzy logic and Neuro Fuzzy approaches for monitoring wall,” In: Design & Technology of Integrated Systems in Nanoscale Era (DTIS), 2012 7th International Conference on. IEEE, 2012. pp. 1-6.

[11] N. Eiamkanitchat, N. Theera-Umpon and S. Auephanwiriyakul, “A Novel Neuro-fuzzy Method for Linguistic Feature Selection and Rule-Based Classification,” In: Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on. IEEE, 2010. pp. 247-252.

[12] N. Eiamkanitchat, N. Theera-Umpon and S. Auephanwiriyakul, “Colon Tumor Microarray Classification Using Neural Network with Feature Selection and Rule-Based Classifica-tion,” In: Zeng Z., Wang J. (eds.). LNEE, Vol. 67, Springer, Berlin Heidelberg, 2010. pp. 363-372.

[13] K. Bache and M. Lichman, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.