A GA-Based Novel RBF Classifier with Class-Dependent Features

Xiuju Fu and *Lipo Wang**
School of Electrical and Electronic Engineering
Nanyang Technological University
Block S2, Nanyang Avenue
Singapore 639798
Email: {p146793114,elpwang}@ntu.edu.sg
http://www.ntu.edu.sg/home/elpwang
*Corresponding author

Abstract - High dimensionality of data sets is a curse to classifiers. We propose to construct a novel radial basis function (RBF) classifier using class-dependent features by genetic algorithms (GA). Since each feature may have different capabilities in discriminating different classes, features should be masked differently for different classes. In our novel RBF classifier, each Gaussian kernel function of the RBF neural network is active for only a subset of patterns which are approximately of the same class. A group of Gaussian kernel functions is generated for each class. In our method, different feature masks are used for different groups of Gaussian kernel functions corresponding to different classes. The feature masks are adjusted by GA. The classification accuracy of the RBF neural network is used as the fitness function. Thus, the dimensionality of a data set is reduced. Simulations show that, with irrelevant features removed for each class, our method can lead to significant improvements on classification accuracy.

I. INTRODUCTION

Many algorithms have been developed for data dimensionality reduction (DDR). In general, techniques for DDR may be classified into two categories: class-independent (features selected are common to all classes) and class-dependent (different feature sets are selected for different classes). In class-independent DDR [6][8][12], all features selected are assumed to play equal roles in discriminating each class from the others.

Class-independent features can be obtained through genetic algorithms (GA) [2][3][5][7][13]. Each chromosome in the population pool represents a feature mask [2][3][7][13]. Assume there are n features in total. n bits are needed in a chromosome to represents the n features. The kth bit of a chromosome indicates the presence or absence of the kth feature, i.e., the feature is presented if the bit is 1, and is absent if the bit is 0. The classification accuracy of a classifier may be used as the fitness function in GA. Fung *et al* [5] proposed to use fuzzy GA (FGA) to select class-independent features. In [2][5], the fitness evaluator is a nearest-neighbor (NN) classifier. Chaikla and Qi [3] also chosed a NN classifier together with multiple correlation as the fitness function of GA to select class-independent features. Raymer *et al* [13] encoded the number of NN classifiers into a chromosome together with the features. In [7], the classification result was determined by a vote of several different classifiers, i.e., the logistic classifier (LOG), the linear discriminant classifier (LDC) and the quadratic discriminant classifier (QDC), etc..

Oh *et al* [10][11] proposed class-dependent feature selection to improve recognition performance. Class-dependent features were selected by considering class separation in conjunction with the recognition rate [11]. Next, Oh *et al* constructed multiple MLP classifiers based on the class-dependent features obtained. For each class, an MLP classifier whose inputs were the features selected for this class was trained individually. Thus, if there are M classes in the data set, M MLP classifiers have to be trained, which is computationally expensive.

In this paper, we propose a novel RBF classifier based on the possibility that a feature may have different capability in discriminating different classes. For different groups of hidden units corresponding to different classes, different feature subsets are selected as inputs. GA is used to search for the optimal feature masks. In other words, we incorporate Oh *et al*'s class-dependent feature selection [10][11] in the RBF classifier. In contrast to Oh *et al* [10][11], only a single such RBF network, rather than

0-7803-7282-4/02/$10.00 ©2002 IEEE

multiple MLPs, are required for a multi-class problem.

This paper is organized as follows. The traditional RBF classifier is introduced briefly in Section II. In Section III, we propose a new RBF classifier with class-dependent feature masks, which are encoded by GA. Experimental results are shown in Section IV. Finally, we conclude the paper in Section V.

II. THE CONVENTIONAL RBF CLASSIFIER

RBF neural networks [1][14][15] are widely used for function approximation, pattern classification, and so on. In an RBF neural network, the activation of a hidden unit is determined by the distance between the input vector and the center vector of the hidden unit. The weights connecting the hidden layer and the output layer can be determined by a linear least square (LLS) method [1], which is fast and free of local minima, in contrast to the MLP neural network.

There are three layers in the RBF neural network, i.e., the input layer, the hidden layer with Gaussian activation functions, and the output layer. The architecture of the conventional RBF neural network is shown in Fig.1. There are M classes in the data set and M output neurons. The m-th output of the network is written as follows:

$$y_m(\mathbf{X}) = \sum_{j=1}^{K} w_{mj}\phi_j(\mathbf{X}) + w_{m0}b_m \quad . \qquad (1)$$

Here $\mathbf{X}$ is the n-dimensional input pattern vector. $\mathbf{X} = \{x_1, x_2, ..., x_k, ..., x_n\}$. $m = 1, 2, ..., M$. K is the number of hidden units. w_{mj} is the weight connecting the j-th hidden unit to the m-th output node. b_m is the bias. w_{m0} is the weight connecting the bias in the m-th output neuron. $\phi_j(\mathbf{X})$ is the activation function of the j-th hidden unit:

$$\phi_j(\mathbf{X}) = e^{\frac{||\mathbf{X}-\mathbf{C_j}||^2}{2\sigma_j^2}} \quad , \qquad (2)$$

where $\mathbf{C_j} = \{c_1, c_2, ..., c_k, ..., c_n\}$ and σ_j are the center and the width for the j-th hidden unit, respectively, which are adjusted during learning.

III. CONSTRUCTING AN NOVEL RBF CLASSIFIER

In this section, we propose a novel RBF classifier and describe its constructing algorithm. We observe that the hidden neurons in an RBF network may be grouped according to classes. That is, if most of the patterns in the cluster represented by a hidden neuron belong to class i, we say that this hidden neuron belongs to the group for

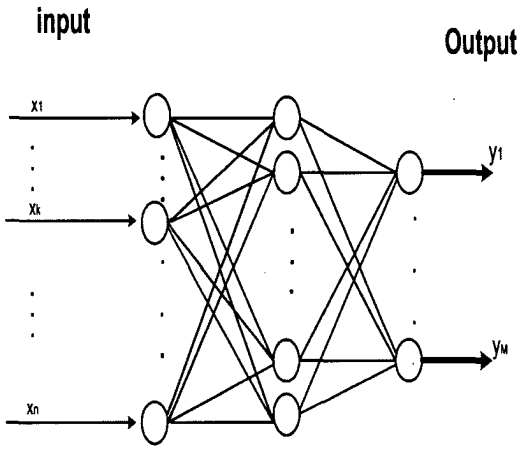

Fig. 1. Architecture of a conventional RBF neural network

class i (Fig. 2). We add a class-dependent feature mask for each group of hidden neurons.

The m-th output of the network is as follows:

$$y_m(\mathbf{X}) = \sum_{i=1}^{M}\sum_{j=1}^{k_i} w_{mj}^i \phi_j^i(\mathbf{X}) + w_{m0}b_m \quad , \qquad (3)$$

where $\phi_j^i(\mathbf{X})$ is the activation function of the j-th hidden unit which serves class i:

$$\phi^i{}_j(\mathbf{X}) = e^{\frac{||\mathbf{X^i}-\mathbf{C_j^i}||^2}{2\sigma_j^{i\,2}}} \quad . \qquad (4)$$

Here $\mathbf{X^i} = \{g_1^i x_1, g_2^i x_2, ..., g_k^i x_k, ..., g_n^i x_n\}$. $\{g_1^i, g_2^i, ..., g_k^i, ..., g_n^i\}$ is the feature mask for class i. $g_k^i = 0, 1$. σ_j^i is the width for the j-th hidden unit of class i and is obtained during training in the presence of the feature masks. $\mathbf{C_j^i} = \{g_1^i c_1, g_2^i c_2, ..., g_k^i c_k, ..., g_n^i c_n\}$.

Finding the centers, widths and the weights connecting hidden nodes to the output is the key to constructing and training the RBF classifier. Both the dimensionality and the distribution of the input patterns affect the number of the hidden units.

The notation used is as follows. δ_0^i is the maximum radius of the receptive field of the Gaussian kernel function for class i. δ_0^i is the standard deviation of the distances from the patterns of class i to the centroid of class i [14][15][1]. θ is the percentage of in-class patterns (the patterns belongs to the current class processed) in a cluster. θ controls the nature of Gaussians [14][15]. For example, a cluster with $\theta = 60\%$ (a "fat" Gaussian, there are 60% in-class patterns in the cluster) can detect global features in the data set that might not be detected by the cluster with a larger θ (a "narrow" Gaussian) [14][15], e.g. , $\theta = 90\%$. V is the training set. α is the changing rate

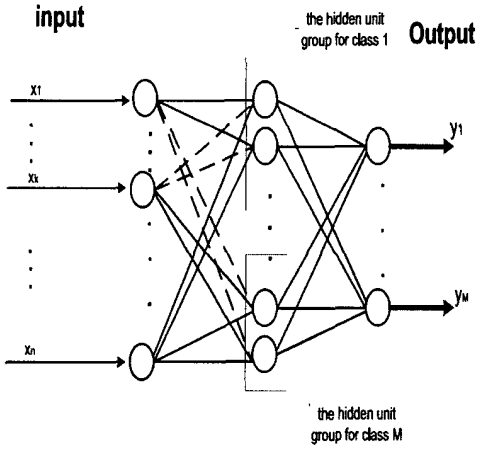

Fig. 2. Architecture of a new RBF neural network. The dashed lines connecting features with a group of hidden units indicate that these features are not input to that group of hidden units.

of radii of clusters. L is the training stage. N_{Least} is the least number of patterns required for a cluster. Q_i is the number of total patterns that belong to class i in the training set. K_s counts the number of trials intending to generate a cluster. K^i is the number of patterns of class i left in V.

The follow is a modified algorithm based on that of Roy *et al* [14][15]:

1. Initializing:

   a) Divide the data set into three parts, i.e., the training data set, the validation data set, and the test data set.

   b) A binary string G with nM bits in length is used to mask features for M classes.

   c) In order to derive the widths of the kernel functions, δ_0^i corresponding to class i is obtained by calculating the standard deviation of the distance from the patterns of class i to the centroid of class i.

   d) Set class labe i=1.

   e) $N_{Least} = \frac{1}{2}Q_i$. $\theta = 95\%$. $L = 0$. $K_s = 0$. $\alpha(L) = 0.1\delta_0^i$.
2. Forming clusters:

   a) Select randomly a pattern from V as an initial center belonging to class i. $K_s = K_s + 1$. If $K_s > 0.5K^i$, $\theta = 0.9\theta$, $N_{Least} = 1/2N_{Least}$, and $K_s = 0$.

   b) $L = L + 1$. If $L > 10$, go to 2(a). Else set $\delta(L) = \delta_0^i - (L-1)\alpha$ and search in V to find all the patterns in $\delta(L)$-neighborhood of the selected center.

   c) Check two conditions. Condition 1 is whether the ratio between in-class patterns and the total patterns in the cluster is greater than θ. Condition 2 is whether the number of patterns in the cluster is greater than N_{Least}. If condition 1 is not met and condition 2 is met, go to 2 (*b*). If both are true, a cluster is formed, and the patterns in the cluster is removed from V. If condition 1 is met and condition 2 is not met, or neither condition 1 or 2 are not met, continue. $L = 0$. $K_s = 0$. Repeat 2 until there is no pattern belonging to class i left in V.
3. $i = i + 1$. If $i < M$, V is set as the initial training set. Go to 1 (*e*).
4. Calculate the center and width of each cluster: the center is the mean pattern of all patterns in the cluster and the width is the standard deviation of these patterns.
5. Obtain weights connecting the hidden layer and the output layer by the LLS method [1].
6. Calculate E_{tr} (the classification error of the training set) and E_v (the classification error of the validation set). E_v is used as the fitness function in GA for evaluating each individual in the population pool (eq.5).

We use the roulette wheel selection to select chromosomes in each generation. In the roulette wheel selection, the selection probability is proportional to each chromosome's fitness. Two-point crossover is used. Two points are randomly located in each of the two parents. The two parts of the parent chromosomes between the two pairs of points are then exchanged to generate new offsprings. The probability of crossover is 80%.

Mutation can prevent the fixation at some particular loci. A locus in the parent chromosome is selected randomly and the bit at the position is replaced (if the original bit is 0, it is replaced by 1, and vice versa). Usually, the mutation rate is relatively small to avoid too much variation. However, at later generations, the number of identical members increases, which leads to a stagnant state. In order to break stagnant states to search for optimal results, we use a dynamic mutation rate, i.e., if the number of identical members in a population exceeds a certain percentage, the mutation rate is increased by a certain amount.

Our fitness function is:

$$F(G) = 1 - E_v(G) \tag{5}$$

where $E_v(G)$ is the classification error rate of the validation data set for chromosome G.

IV. EXPERIMENTAL RESULTS

Glass and Thyroid data sets from the UCI Repository of Machine Learning Databases [9] are used in this paper to test our algorithm.

The GA parameters are as follows. There are $4n = 36$ (n is the number of features) chromosomes for Glass data set, and 20 chromosomes for Thyroid data set. The initial mutation rate is 40%. If the number of identical members in a population exceeds 25%, the mutation rate is increased by 1%. The number of elite chromosomes, which remain unchanged and live from one generation to the next, is 2. The number of generations is 50.

A. Glass Data Set

Glass data set contains 214 cases. There are 9 attributes and 6 classes in Glass data set. 114 patterns are for training, 50 for validation, 50 for testing.

The error rates are 16.41% for training, 20.93% for validation, and 23.26% for testing. For class 1, 4 features are involved for classifying it from other classes, i.e., 4 features are used as the inputs to the hidden units of class 1 (Table I). 7, 5, 4, 5, 3 features are used as inputs to the hidden units of classes 2, 3, 4, 5, 6, respectively. Thus, the average number of features used for each class is $(4 + 7 + 5 + 4 + 5 + 3)/6 = 28/6 = 4.7$, compared to the original 9 features. Our experimental result is

TABLE I
FEATURE MASK FOR GLASS.

Classes	Feature masks
Class 1	0 0 1 1 0 1 0 1 0
Class 2	1 1 1 1 1 0 1 0 1
Class 3	1 1 0 0 1 1 0 1 0
Class 4	0 1 1 0 1 0 1 0 0
Class 5	0 0 0 1 1 1 0 1 1
Class 6	1 1 0 1 0 0 0 0 0

compared with other methods' results [7] in Table II. In the first line of Table II, the classification result of the LOG classifier [7] using all the features (no feature selection) is shown. By the sequential backward selection (SBS) [4][7], 6.9 class-independent features on average are obtained for all classes. The LOG classification result [7] based on selecting class-independent features using GA is shown in the third line of Table II. There are two versions of another algorithm involving multiple classifiers [7], i.e., the logistic classifier (LOG), the linear discriminant classifier (LDC) and the quadratic discriminant classifier (QDC), etc.. In multiple classifiers

TABLE II
COMPARISON WITHE OTHER METHODS FOR GLASS DATA SET. (NA: NOT AVAILABLE).

Method	Training error	Testing error	Feature number
LOG without feature selection	28.14%	39.54%	9
SBS	27.62%	37.39%	6.9
LOG with class-independent feature selection	27.52%	40.39%	7.7
Multiple classifiers Version 1	27.62%	39.26%	NA
Multiple classifiers Version 2	25.35%	34.55%	NA
Our method	16.41%	23.26%	4.7

(version 1) [7], GA was used for class-independent feature selection and only the features were encoded in GA. The fitness of each chromosome was determined by a vote of the classifiers. In multiple classifiers (version 2), the classifier types were also encoded in GA, i.e., each individual classifier can be chosen from the three classifiers mentioned above. The comparison shows that a better classification accuracy with a fewer number of features is obtained by our method.

B. Thyroid Data Set

There are 5 attributes and 215 patterns in Thyroid data set. 115 patterns are for training, 50 for validation, and 50 for testing.

With class-dependent features, the classification error rates are 2.84% for training, 2.33% for validation, and 4.65% for testing. Without feature masks, the classification error rates are: 3.88% for training, 3.88% for validation, and 4.65% for testing.

It is shown in the feature masks (Table III) that feature 1 does not play any role on discriminating the classes. For class 3, feature 2 can discriminate it from other classes. Feature 2 and 3 are used to classify class 2 from other classes. Feature 2, 3, 4, 5 is used for discriminate class 1 from other classes. Thus, the average number of features used for each class is $(1+2+4)/3 = 7/3 = 2.33$, compared to the original 5 features.

TABLE III
FEATURE MASK FOR THYROID DATA SET.

Classes	Feature masks
Class 1	0 1 1 1 1
Class 2	0 1 1 0 0
Class 3	0 1 0 0 0

V. CONCLUSIONS

In this paper, we have proposed a class-dependent feature selection method based on a novel RBF classifier and GA. The feature subset is selected for each class individually based on its ability in discriminating the class with other classes. Glass and Thyroid data sets are used to test the algorithm. Experimental results show that the algorithm proposed are effective in reducing the number of feature input and improving classification accuracy simultaneously.

DDR is often the first step for data mining tasks. The class-dependent feature selection results obtained above provide a new direction for analyzing the relationships between features and classes. The reduction in dimensionality can lead to compact rules in rule extraction task. Extracting rules based on the classification results obtained above will be our future work.

References

[1] C. M. Bishop, *Neural network for pattern recognition*, Oxford University Press, New York, 1995.

[2] F. Z. Brill, D. E. Brown, and W. N. Martin, "Fast generic selection of features for neural network classifiers", *IEEE Transactions on Neural Networks*, vol. 3, no. 2, pp. 324 - 328, March 1992.

[3] N. Chaikla and Y. L. Qi, "Genetic algorithms in feature selection", *1999 IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, pp. 538 - 540, 1999.

[4] P. A. Devijver and J. Kittler, *Pattern recognition: a statistical approach*, Prentice-Hall International, Inc. London, 1982.

[5] G. S. K. Fung, J. N. K. Liu, K. H. Chan, and R. W. H. Lau, "Fuzzy genetic algorithm approach to feature selection problem", *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, vol. 1, pp. 441 - 446, 1997.

[6] N. Kambhatla and T. K. Leen, "Fast non-linear dimension reduction", *IEEE International Conference on Neural Network*, vol. 3, pp.1213 - 1218, 1993.

[7] L. I. Kuncheva and L. C. Jain, "Designing classifier fusion systems by genetic algorithms", *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 4, pp. 327 - 336, Nov. 2000.

[8] R. Lotlikar and R. Kothari, "Bayes-optimality motivated linear and multilayered perceptron-based dimensionality reduction", *IEEE Transactions on Neural Networks*, vol. 11, no. 2, pp. 452 - 463, March, 2000.

[9] P. M. Murphy, & D. W. Aha, (1994). *UCI Repository of machine learning databases* [http://www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[10] Il-Seok Oh, Jin-Seon Lee, and C. Y. Suen, "Using class separation for feature analysis and combination of class-dependent features", *Fourteenth International Conference on Pattern Recognition*, vol. 1, pp. 453 - 455, 1998.

[11] Il-Seok Oh, Jin-Seon Lee, and C. Y. Suen, "Analysis of class separation and combination of class-dependent features for handwriting recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 1089 - 1094, Oct. 1999.

[12] P. Pudil and J. Hovovicova, "Novel methods for subset selection with respect to problem knowledge", *IEEE Intelligent Systems [see also IEEE Expert]*, vol. 13 no. 2, pp. 66 - 74, March-April, 1998.

[13] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality reduction using genetic algorithms", *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 2, pp. 164 - 171, July, 2000.

[14] Asim Roy, Sandeep Govil, and Raymond Miranda, "An algorithm to generate radial basis function (RBF)-like nets for classification problems", *Neural networks*, vol. 8, no. 2, pp. 179 - 201, 1995.

[15] Asim Roy, Sandeep Govil, and Raymond Miranda, "A neural-network learning theory and a polynomial time RBF algorithm", *IEEE Transactions on neural network*, vol. 8, no. 6, pp. 1301 - 1313, November, 1997.