J Electr Eng Technol Vol. 9, No. 6: 2098-2106, 2014
http://dx.doi.org/10.5370/JEET.2014.9.6.2098

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

Polynomial Fuzzy Radial Basis Function Neural Network Classifiers Realized with the Aid of Boundary Area Decision

Seok-Beom Roh* and Sung-Kwun Oh†

Abstract – In the area of clustering, there are numerous approaches to construct clusters in the input space. For regression problem, when forming clusters being a part of the overall model, the relationships between the input space and the output space are essential and have to be taken into consideration. Conditional Fuzzy C-Means (c-FCM) clustering offers an opportunity to analyze the structure in the input space with the mechanism of supervision implied by the distribution of data present in the output space. However, like other clustering methods, c-FCM focuses on the distribution of the data. In this paper, we introduce a new method, which by making use of the ambiguity index focuses on the boundaries of the clusters whose determination is essential to the quality of the ensuing classification procedures. The introduced design is illustrated with the aid of numeric examples that provide a detailed insight into the performance of the fuzzy classifiers and quantify several essentials design aspects.

Keywords: Polynomial fuzzy radial basis function neural network, Conditional fuzzy clustering, Pattern classification, Imbalanced data

1. Introduction

Radial basis function (RBF) networks have been widely studied and applied to various tasks of regression and classification problem, cf. [16, 19]. Since the concepts of RBF neural networks were introduced in the literature [20], there have been a number of interesting and useful generalizations of the generic topology of these networks and their learning methods, cf. [1, 4, 12, 14]. The visible feature of RBF neural networks comes with their fast two-phase training method. During this learning process, the values of the parameters of the radial basis functions are determined independently from the weight values of the output layer. Typically, the parameters of the basis functions (referred to as receptive fields) are estimated by some relatively fast and general methods of unsupervised learning applied to input data. After the basis functions have been determined, the output layer's weights are obtained as the least-squares solution to a system of linear equations (e.g., by using the Moore-Penrose pseudo-inverse [2]). Compared to the nonlinear optimization that is usually considered in the training of neural networks, this two-stage method is usually much faster and can help avoid local minima and eliminate difficulties with the convergence of the overall learning process [13].

Let us look in more detail at this two-step design of RBF neural networks by highlighting the diversity of the optimization tools available there.

(a) Optimization of the hidden layer: We encounter a significant variety of radial basis functions being used and face with diverse ways of their development. Discussed are such typical forms of RBFs as Gaussian functions. Other analytical versions of such functions are also available; see [5]. An alternative way of dealing with the formation of the RBFs (receptive fields) is to exploit various clustering techniques including its commonly encountered representatives such as K-means and Fuzzy C-Means (FCM) [19]. Furthermore, the optimization method such as Particle Swarm Optimization can be used to position the RBFs in the input space, refer to [21].
(b) Optimization of the output layer: For learning scheme for the linear neuron located at the output layer of the network, gradient-based methods and Expectation Maximization (EM)-based training method are in common usage, see [9, 11].

Let us consider a way in how to locate the receptive fields of RBF neural networks (i.e., how to analyze and describe the input space which is inherently related with the output space through some unknown function whenever the output space is the real numbers space present in case of regression problems or the space of integers used in classification tasks). As noted above, various unsupervised clustering methods such as K-means and Fuzzy C-Means have been proposed to construct receptive fields. In particular, for regression problems, Pedrycz [18] has pointed at a certain drawback of the original objective function based clustering techniques

† Corresponding Author: Dept. of Electrical Engineering, The University of Suwon, Korea. (ohsk@suwon.ac.kr)
* Dept. of Electronic and Information Engineering, Wonkwang University, Korea. (nado@wonkwang.ac.kr)
Received: January 27, 2014; Accepted: April 24, 2014

such as Fuzzy C-Means clustering. This shortcoming, which is commonly encountered when using the clustering methods based on the minimization of the objective function to form linguistic terms of a fuzzy model over the input space, is that all of those terms are formed in a completely unsupervised manner even though there is some component of supervision available which comes in the form of dependent (output) variables. To alleviate this shortcoming and take into account information about the output space, the Conditional Fuzzy C-means (c-FCM) clustering has been proposed. Given that the information about the output is used in the method, it brings some component of supervision to the clustering process.

In this study, we develop a concept of RBF neural networks based on the supervisory clustering (supervision-augmented clustering) which relates with supervisory clustering realized for regression problems. Furthermore, when dealing with classification problems, the supervisory clustering has to be activated within the boundary area occupied by patterns to be classified. We define the boundary area as a certain region of the input space where the data (patterns) belonging to different classes are located. Given a mixture of data coming from different classes or associated with a substantial variety of output's values, the boundary region can be regarded as a source of useful discriminatory information. In contrast, the regions of the input space associated within the core of each class (where by the core we mean a region of the input space being predominantly occupied by patterns belonging to the same class) might be a limited source of discriminatory information.

In order to activate the supervised clustering within the boundary area, we describe this area by using several linguistic terms (quantified in terms of fuzzy sets). This approach is legitimate considering that fuzzy sets are naturally geared to describe concepts (here classes) exhibiting overlap with elements belonging to other classes. After determining the boundary area, we invoke supervisory clustering to analyze the structure of the space. The performance of the proposed classifier is contrasted with the results produced by polynomial Fuzzy Radial Basis Function Neural Networks (pFRBF NNs). To show the classification abilities of the proposed classifier preferred to the various types of classifiers, we compare the generalization ability of the proposed classifier with the well-known classifiers.

This study is organized as follows. In Section 2, we review the generic architecture of the generic RBF NNs and the extended RBF NNs. Next, in Section 3, we propose and elaborate on the pFRBF NNs classifiers focused on the boundary decision area and conditional fuzzy clustering. Extensive experimental studies are covered in Section 5 while Section 6 offers some concluding comments.

2. Architecture of the Extended pRBFNNs

Several researches have said that the generic pFRBF NNs exhibit some advantages including global optimal approximation and classification capabilities as well as rapid convergence of the underlying learning procedures, see [6, 8]. The generic topology of pFRBF NNs is depicted in Fig. 1.

In Fig. 1, Γ_i, i=1, 2,…, c denotes receptive fields (radial basis functions), while "m" stands for the number of the input variables. The output of the generic pFRBF NN comes as a linear combination of the outputs ($\Gamma(\mathbf{x})$) of the corresponding nodes at the hidden layer with the connection weights $w_1, w_2, \cdots, w_c$ as shown below

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{c} w_i \cdot \Gamma_i(\mathbf{x}) \tag{1}$$

Where $\mathbf{x} = [x_1 \; x_2 \; \cdots \; x_m] \in \Re^m$ and $\Gamma_i(\mathbf{x})$ is the activation level of the i-th node present at the hidden layer.

Generally, the Gaussian type pFRBFs are used as receptive fields

$$\Gamma_i(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{v}_i\|^2}{2\sigma_i^2} \right) \tag{2}$$

where $\mathbf{v}_i$ and σ_i are the apex (center) and the spread of the i[th] receptive field, respectively.

There are two major differences between the extended pFRBF NNs and the generic version of pFRBF NNs. The first one concerns the type of the underlying receptive fields. In the extended pFRBF NNs, the prototypes of the receptive fields (i.e., the nodes of the hidden layer) are determined by running fuzzy clustering. The output of each node in the hidden layer is an activation level of the corresponding linguistic term (fuzzy set)

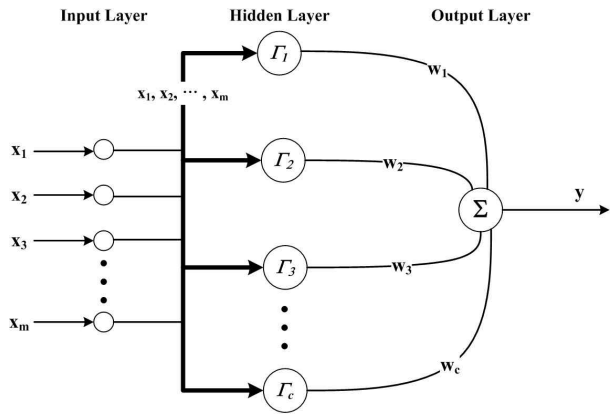


Fig. 1. General architecture of the generic pFRBF Neural Networks

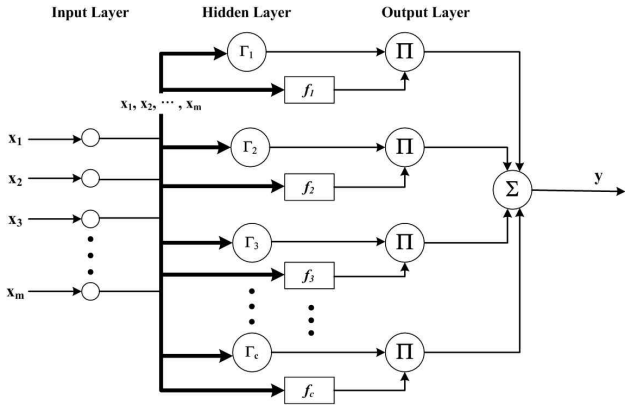


Fig. 2. Architecture of the extended pFRBF Neural Networks

$$RF_i = \Gamma_i(\mathbf{x}) = \frac{1}{\sum_{j=1}^{c}\left(\frac{\|\mathbf{x}-\mathbf{v}_i\|^2}{\|\mathbf{x}-\mathbf{v}_j\|^2}\right)} \tag{3}$$

The second difference arises in terms of the type of the connection (weights) between the hidden layer and output layer. In the extended pFRBF NNs, we use linear functions or the 2nd order polynomials rather than confining ourselves to some fixed numeric values. The architecture of the extended pFRBF NN and the type of connection weights considered above is shown in Fig. 2.

In Fig. 2, f_i denotes the connection (weight) between the ith node of hidden layer and the node in the output layer. The connection f_i is expressed as a linear function or the 2nd order polynomial. More specifically, we have

$$f_i(\mathbf{x}_k) = a_{i0} + \sum_{j=1}^{m} a_{ij}x_j = [1 \ \mathbf{x}_k]\mathbf{a}_i \text{ (linear function)} \tag{4}$$

Here, $\mathbf{a}_i = [a_{i0} \ a_{i1} \ \cdots \ a_{im}]^T \in \Re^{(m+1)}$

$$\begin{aligned} f_i(\mathbf{x}_k) &= a_{i0} + \sum_{j=1}^{m} a_{ij}x_j + \sum_{j=1}^{m}\sum_{l=j}^{m} a_{ijl}x_j x_l \\ &= [1 \,|\, \mathbf{x}_k \,|\, x_{k1}x_{k1} \ x_{k1}x_{k2} \ \cdots \ x_{km}x_{km}]\mathbf{a}_i \\ &\quad (2^{\text{nd}} \text{ order polynomial}) \end{aligned} \tag{5}$$

Here,

$$\mathbf{a}_i = [a_{i0} \ a_{i1} \ \cdots \ a_{im} \ a_{i11} \ \cdots \ a_{imm}] \in \Re^{\frac{(m+2)(m+1)}{2}}$$

The activation level of each node in the hidden layer is determined using (3). The normalized activation level u_{ik} follows the expression

$$u_{ik} = u_i(\mathbf{x}_k) = \frac{\Gamma_i(\mathbf{x}_k)}{\sum_{j=1}^{c}\Gamma_j(\mathbf{x}_k)} \tag{6}$$

The following relationship holds

$$\sum_{i=1}^{c} u_i(\mathbf{x}_k) = \sum_{i=1}^{c}\frac{\Gamma_i(\mathbf{x}_k)}{\sum_{j=1}^{c}\Gamma_j(\mathbf{x}_k)} = 1 \tag{7}$$

For the output node we obtain

$$\hat{y}_k = \sum_{i=1}^{c} u_{ik} f_i(\mathbf{x}_k) = \frac{\sum_{i=1}^{c}\Gamma_i(\mathbf{x}_k)\cdot f_i(\mathbf{x}_k)}{\sum_{i=1}^{c}\Gamma_i(\mathbf{x}_k)} \tag{8}$$

3. The Development of The pFRBFNN Classifier Activated Within the Boundary Area

When we consider a two class problem, as elaborated on in the introduction, we use the extended pFRBF NNs to be used as the primary classifier whose receptive fields are constructed by running supervised clustering (i.e., the conditional fuzzy C-Means).

The basic conjecture coming with the proposed classifier is that more *ambiguous* information is present within the boundary area than with the core area (generally speaking, the core area has homogeneous patterns belonging to the same class whereas the boundary area typically embraces patterns belonging to several classes).

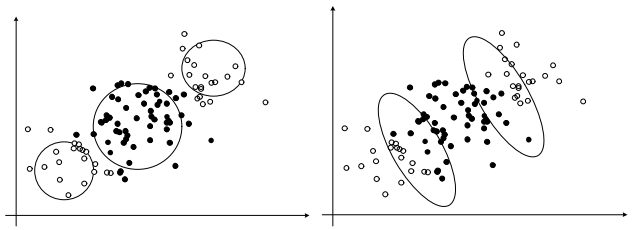
The boundary surface is formed within boundary area. In this paper, the boundary surface (area) for each class is determined by using the extended pFRBF NNs as already presented in Section II. The output of extended pFRBF NNs is aggregated through a certain linear combination of the local models, which describe the relationship between the input variables and the output variables present within the related local areas. The local model (i.e., the linear function or the 2nd order polynomial) of pFRBF NNs defines the local boundary surface, which is formed within the local area defined by the receptive field.

We anticipate that the improvement of classification performance becomes associated with the use of the receptive fields that are positioned within the boundary area.

3.1 Defining the boundary area

Let us recall that the boundary area pertains to the region in the input space in which we encounter patterns belonging to different classes. In contrast, the core area (region) is highly homogeneous where there are data belonging to the same class. Fig. 3 illustrates some core and boundary areas formed for the two-class data.

In order to define the boundary areas by using linguistic terms, the data patterns involved in each class are previously analyzed by the Possibilistic C-Means (PCM) clustering. As far as data set is concerned, we consider a



(a) Core area (b) Boundary area

Fig. 3. Examples of core and boundary areas

finite set of "n" input-output data coming in the form of the ordered pairs $\{\mathbf{x}_k, g_k\}$, $k = 1, 2, \cdots, n$, $\mathbf{x}_k \in \Re^m$, while $g_k \in \{1, 2, \cdots, l\}$, l is the number of classes. The output variable g_k is the class label. Denote by L_i the set of indices of the data pattern involved in the i-th class.

$$L_i = \{k \mid g_k = i, k = 1, 2, \cdots, n\} \tag{9}$$

The original FCM uses the probabilistic constraint meaning that the membership grades for the same data sum up to one. While this is useful in forming the partition, the membership values resulting from the FCM and the related methods, however, may not always correspond to the intuitive concept of degree of belongingness, compatibility or typicality (commonality) as noted in the literature. Krishnapuram and Keller relaxed this constraint and introduced possibilistic clustering (PCM) by minimizing the following objective function

$$J(U,V) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{p} \left\|\mathbf{x}_k - \mathbf{v}_i\right\|^2 + \sum_{i=1}^{c} \eta_i \sum_{k=1}^{n} (1-u_{ik})^p \tag{10}$$

where η_i is a certain positive number, and "p" is a fuzzification coefficient that should be determined as any real number greater than 1 which is the same parameter as that is used in the ordinary FCM.

The first term requires that the distances from data points to the prototypes be as low as possible while the second term forces the values of u_{ik} to be as large as possible, thus avoiding running into a trivial solution. It is recommended to select η_i as discussed in [10], that is

$$\eta_i = K \frac{\sum_{j=1}^{n} \left(u_{ij}\right)^p \left\|\mathbf{x}_j - \mathbf{v}_i\right\|}{\sum_{j=1}^{n} \left(u_{ij}\right)^p} \tag{11}$$

Typically, the value of K is chosen to be equal to 1. The update of the prototypes is realized in the same way as this has been done in the FCM algorithm,

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} \left(u_{ik}\right)^p \cdot \mathbf{x}_k}{\sum_{k=1}^{n} \left(u_{ik}\right)^p} \tag{12}$$

The membership degree (partition matrix) in the PCM is calculated as follows

$$u_{ik} = \frac{1}{1 + \left(\frac{\left\|\mathbf{x}_k - \mathbf{v}_i\right\|^2}{\eta_i} \right)^{1/(p-1)}} \tag{13}$$

We determine the prototypes and the activation levels for each class separately. The prototypes of class "j" (i.e. $g_k = j$) is calculated as follows

$$\mathbf{v}_i^j = \frac{\sum_{k=1}^{n_j} \left(u_{ik}\right)^p \cdot \mathbf{x}_{L_j\{k\}}}{\sum_{k=1}^{n} \left(u_{ik}\right)^p} \tag{14}$$

$\mathbf{v}_i^j$ is the prototype of the i-th cluster of the j-class, n_j is the number of elements of the index set L_j, and $L_{j\{k\}}$ means the k-th element of the index set L_j.

The activation level of the i-th cluster for the j-class is calculated as follows.

$$u_{ik}^j = \frac{1}{1 + \left(\frac{\left\|\mathbf{x}_k - \mathbf{v}_i^j\right\|^2}{\eta_i^j} \right)^{1/(p-1)}} \tag{15}$$

Where $\eta_i^j = K \dfrac{\sum_{k=1}^{n_j} \left(u_{ik}^j\right)^p \left\|\mathbf{x}_{L_j\{k\}} - \mathbf{v}_i^j\right\|}{\sum_{k=1}^{n_j} \left(u_{ik}^j\right)^p}$ is the specific η_i available in the j-th class.

Note that these expressions are the modified versions of (12) and (13).

The higher the activation levels (15) are, the more visibly the data is involved in the core area of the corresponding class.

After calculating the activation levels and prototypes for all classes, we define the boundary area as follows.

$$B(\mathbf{x}_k) = S\left(T\left(u_{ik}^1, u_{jk}^2\right)\right), \quad i = 1, \cdots, C \text{ and } \quad j = 1, \cdots, C \tag{16}$$

Here T stands for some t-norm and S denotes a certain t-conorm (s-norm). In this study, the t-norm is realized as the minimum operator and the t-conorm is specified as the probabilistic sum.

u_{ik}^1 and u_{jk}^2 denote the activation levels of the i-th cluster of "1" class and the j-th cluster of "2" class, respectively. As shown in Fig. 4, with 2 classes where each class is composed of 2 clusters, the boundary area is defined in the following form

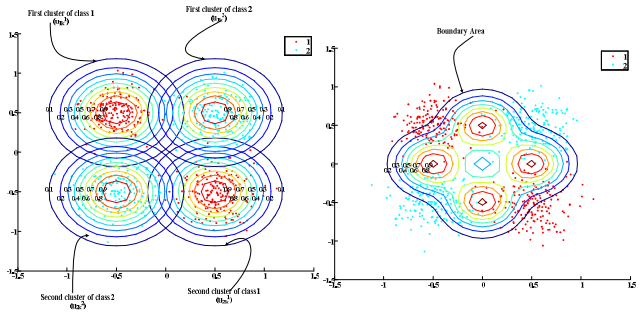


Fig. 4. Examples of the Boundary Area associated with the corresponding values of α-cuts of the fuzzy clusters (membership functions)

$$B(\mathbf{x}_k) = S\left(T\left(u_{1k}^1, u_{1k}^2\right), T\left(u_{1k}^1, u_{2k}^2\right), T\left(u_{2k}^1, u_{1k}^2\right), T\left(u_{2k}^1, u_{2k}^2\right)\right) \tag{17}$$

3.2 Conditional fuzzy C-Means clustering within boundary area

The idea of Conditional Fuzzy C-Means (c-FCM, for short) clustering proposed in [18] was applied to the design of pFRBF neural networks as presented in [19]. To elaborate on the essence of the method, let us consider a set of patterns $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, $\mathbf{x}_k \in \Re^m$ (where m stands for the dimensionality of the input space) along with an auxiliary information granule, which is defined as the boundary area. Each element of X is then associated with the auxiliary information granule (fuzzy set) B given by (16).

In conditional clustering, the data pattern $\mathbf{x}_k$ is clustered by taking into consideration the conditions (auxiliary information expressed in the form given by $B(\mathbf{x}_1)$, $B(\mathbf{x}_2), \cdots, B(\mathbf{x}_n)$) based on some linguistic term expressed as a fuzzy set B ($B: \Re \rightarrow [0,1]$). The objective function used in the conditional fuzzy clustering is the same as the one used in the FCM, namely

$$J = \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^p \cdot \left\|\mathbf{x}_k - \mathbf{v}_i\right\|^2 \tag{18}$$

where J is the objective function, u_{ik} is the activation level associated with the linguistic term B defining the boundary area, $\mathbf{v}_i$ is the i^{th} cluster and c is the number of rules (clusters) formed for this context. The difference between the FCM and c-FCM comes in the form of the constraint imposed on the partition matrix where we now have

$$\sum_{i=1}^{c} u_{ik} = B(\mathbf{x}_k) \tag{19}$$

Here, $B(\mathbf{x}_k)$ is the linguistic term (fuzzy set) which means the activation level how much the input data $\mathbf{x}_k$ is involved in the boundary area. Now the optimization problem is formulated in the following form

$$\min_{\mathbf{U},\mathbf{v}} J \quad \text{subject to} \quad \sum_{i=1}^{c} u_{ik} = B(\mathbf{x}_k) \tag{20}$$

The iterative optimization scheme is governed by the two update formulas using which we successively modify the partition matrix and the prototypes

$$u_{ik} = \frac{B(\mathbf{x}_k)}{\sum_{j=1}^{C}\left(\frac{\|\mathbf{x}_k - \mathbf{v}_i\|}{\|\mathbf{x}_k - \mathbf{v}_j\|}\right)^{2/(p-1)}} \tag{21}$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n}(u_{ik})^p \cdot \mathbf{x}_k}{\sum_{k=1}^{n}(u_{ik})^p} \tag{22}$$

3.3 pRBFNNs classifier- The use of conditional fuzzy C-Means clustering and a focus on the boundary area

In what follows, we propose the pFRBF NNs classifier developed by the c-FCM clustering supervised by the linguistic term, which specifies the boundary area.

As mentioned earlier, we assume that in order to improve the classification performance one has to locate the pFRBFs within the boundary area. pFRBF NNs is composed of the linear combination of the local models, which are defined on the local areas (receptive fields). In this way, the pFRBF NNs classifier can be regarded as a linear combination of the local boundary surfaces.

The local models of the pFRBF NNs are activated within

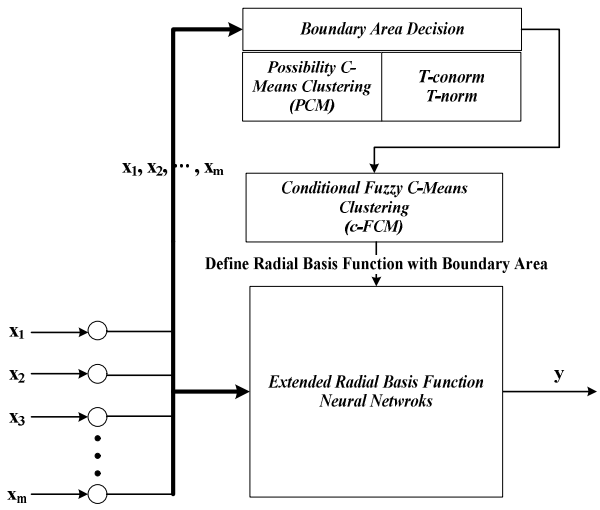


Fig. 5. The overall development of the proposed classifier based on the extended pFRBF NNs, boundary area decision, and c-FCM

the receptive fields (pFRBFs). Therefore, the pFRBFs located within the boundary area have the potential to form the "sound" boundary surface. Fig. 5 shows an overall development process of the proposed classifier.

As shown in Fig. 2, the output of the proposed pFRBF NNs classifier comes as the linear combination of the connection weights such as ($f_1, f_2, \cdots, f_c$) with the activation levels of each node of the hidden layer ($\Gamma_1, \Gamma_2, \cdots, \Gamma_c$). The way to calculate the output of the network of the proposed classifier is similar to the output of the extended pFRBF NNs. However, the activation levels of each pFRBF of the proposed model are described by using (21) which is quite different from the description provided by (2) and (6).

To estimate the connections we use the orthogonal least square method and the weighted least square estimation method. Proceeding with the optimization details, the objective function of Least Square Estimation (LSE) reads as follows

$$J = \sum_{i=1}^{n} (g_i - \sum_{j}^{c} u_{ji} f_j(\mathbf{x}_i))^2 = (\mathbf{G} - \boldsymbol{\Theta}\mathbf{a})^T (\mathbf{G} - \boldsymbol{\Theta}\mathbf{a}) \quad (23)$$

where $f_i(\mathbf{x}) = a_{i0} + \sum_{j=1}^{m} a_{ij} x_j$,

$$\mathbf{G} = \begin{bmatrix} g_1 & g_2 & \cdots & g_n \end{bmatrix}^T,$$

$$\boldsymbol{\Theta} = \begin{bmatrix} u_{11} & u_{11}x_{11} & \cdots & u_{11}x_{m1} & u_{12} & u_{12}x_{11} & \cdots & u_{1c}x_{m1} \\ u_{21} & u_{21}x_{12} & \cdots & u_{21}x_{m2} & u_{22} & u_{22}x_{12} & \cdots & u_{2c}x_{m2} \\ & & & & & & & \\ u_{n1} & u_{n1}x_{1n} & \cdots & u_{n1}x_{mn} & u_{n2} & u_{n2}x_{1n} & \cdots & u_{nc}x_{mn} \end{bmatrix},$$

$$\mathbf{a} = \begin{bmatrix} a_{10} & a_{11} & \cdots & a_{1m} & a_{20} & a_{21} & \cdots & a_{2m} & \cdots & a_{cm} \end{bmatrix}^T \in \Re^{c(m+1)}$$

The optimal values of the coefficients are expressed in a well-known manner

$$\mathbf{a} = (\boldsymbol{\Theta}^T \boldsymbol{\Theta})^{-1} (\boldsymbol{\Theta}^T \mathbf{G}) \quad (24)$$

When we use the weighted LSE to estimate the coefficients of local models, we assume that each data patterns comes with its priority and data patterns with high priority significantly affect the estimation process whereas data with low priority participate to a limited degree and can be almost neglected. The activation levels of the linguistic variable defining the boundary area can be considered as the priority index. As said earlier, we emphasize the data positioned within the boundary area.

Unlike the conventional LSE, the objective function of the weighted LSE is defined as follows

$$J = \sum_{i=1}^{n} \left(\left(B(\mathbf{x}_i) \right)^q \left(g_i - \sum_{j=1}^{c} u_{ji} f_j(\mathbf{x}_i) \right)^2 \right)$$
$$= \left(\mathbf{G} - \boldsymbol{\Theta}\mathbf{a} \right)^T \mathbf{D}^q \left(\mathbf{G} - \boldsymbol{\Theta}\mathbf{a} \right) \quad (25)$$

Where, $\mathbf{D} = \begin{bmatrix} B(\mathbf{x}_1) & & & 0 \\ & B(\mathbf{x}_2) & & \\ & & \ddots & \\ 0 & & & B(\mathbf{x}_n) \end{bmatrix}$.

In the above expression, q denotes the linguistic modifier of the activation level of the boundary area. If the values of q get higher than 1, we arrive at higher specificity of the underlying linguistic information while an opposite effect becomes present when dealing with the lower values of q [3]. Note that the diagonal partition matrix $\mathbf{D}$ is the reduced matrix, which is composed of the activation levels of all data pairs to the linguistic term B as the diagonal elements.

The optimal values of the coefficients by using the weighted LSE are expressed in a well-known manner.

$$\mathbf{a} = \left(\boldsymbol{\Theta}^T \mathbf{D}^{q/2} \boldsymbol{\Theta} \right)^{-1} \left(\boldsymbol{\Theta}^T \mathbf{D}^{q/2} \mathbf{Y} \right) \quad (26)$$

The final output of the pFRBF NNs comes in the form

$$\hat{\mathbf{Y}} = \boldsymbol{\Theta}\mathbf{a} \quad (27)$$

The estimated class label is calculated by using the decision rule

$$\hat{g}_i = \begin{cases} 0, & if\ \hat{y}_i < 0 \\ 1, & f\ \hat{y}_i \geq 0 \end{cases} \quad (28)$$

4. Experimental Study

In order to evaluate and quantify the classification effectiveness of the proposed classifier, the proposed classifier is experimented with by making use of a series of numeric data such as two synthetic datasets and several Machine Learning datasets (http://www.ics.uci.edu/~mlearn/MLRepository.html). In the assessment of the performance of the classifiers, we use the error rate of the resulting classifier.

We investigate and report the results of each experiment in terms of the mean and the standard deviation of the performance index. We consider some predefined values of the parameters of the network whose values are summarized in Table 1. The choice of these particular numeric values has been motivated by the need to come up

Table 1. Selected Numeric Values of the Parameters of the Proposed Model

Parameter	Value
Polynomial order (O)	0 (constant), 1 (linear), or 2 (quadratic)
Number of pFRBFs (c)	2 ~ 10
Fuzzification Coefficient (p)	In the range of 1.2~3.0 varying with step of 0.2
Linguistic Modifier (q)	1.0 or 2.0

with a possibility to investigate of the performance of the model in a fairly comprehensive range of scenarios.

In what follows, we report on several experiments dealing with some machine learning data sets (http://www.ics.uci.edu/~mlearn/MLRepository.html). For simplicity, we deal with two class- problems (the classifier can be extended to deal with more than two classes). The experiments were repeated 10 times using a random split of data into 70%-30% training and testing subsets. Table 2 contrasts the classification error of the proposed classifier with other well-known methods known in the literature [17]. In this experiments, the generic type basic neural networks (NNs), principal component analysis (PCA) and linear discriminant analysis (LDA) are used. Support vector machine (SVM) is available in a MATLAB toolbox, see http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox/. For the decision tree methods, the code of C4.5 trees was coming from the Classification Toolbox of MATLAB (http://www.yom-tov.info/cgi-bin/list_uploaded_files.pl) and the decision trees used some functions coming from the Statistics Toolbox of MATLAB.

Table 3 shows the comparison between the proposed classifier and the classification methods based on the boundary analysis. In this experiments, we use 10 fold cross validation to evaluate the classification abilities and the final correct classification ratio is given in terms of its average and the standard deviation. From the results in Table VI, we can see that the proposed classifier is better than the LBDA based classifiers in terms of the classification abilities achieving higher classification rates.

5. Conclusion

In this paper, we proposed the new design methodology of polynomial fuzzy radial basis function neural networks for the classification problem. Unlike the usual design method of RBFs, the proposed design method concentrate on a detailed description of the boundary regions in the feature space. The learning algorithm used to in the development of the conclusion part of the rules takes advantage of the linear discriminant analysis. To evaluate the proposed model for classification problem, we completed several experiments using 2-dimensional synthetic datasets and a number of machine learning datasets.

Table 2. Results of comparative analysis (The best results are shown in boldface)

Classifiers		Pima	Heart	WDBC	WPBC	Ionosphere	Sonar	German
Proposed model	LSE O=1	21.83±1.7	13.46±2.2	2.98±1.4	18.97±4.4	11.14±3.3	22.42±5.4	22.03±1.8
	LSE O=2	21.22±1.8	13.83±2.5	3.80±1.3	24.66±5.5	12.19±3.4	26.29±5.8	22.33±1.9
	WLSE q=1, O=1	21.04±1.8	13.46±4.0	3.04±1.3	**17.93±2.3**	11.24±3.0	15.32±5.3	22.23±1.7
	WLSE q=1, O=2	21.22±1.8	13.58±3.0	3.57±0.7	24.14±4.6	10.38±2.4	**13.39±3.7**	**21.63±2.1**
	WLSE q=2, O=1	**20.87±2.0**	13.95±3.1	**2.75±1.0**	18.28±4.5	11.14±3.7	14.35±5.3	21.93±1.4
	WLSE q=2, O=2	20.91±2.3	**13.21±2.3**	3.63±1.4	25.52±5.7	10.00±2.6	13.55±4.4	21.87±1.4
RBF NNs	O=1	21.04±1.9	15.19±3.0	3.04±0.8	19.66±2.7	11.24±2.0	24.03±6.4	22.4±1.2
	O=2	21.30±1.9	15.06±2.8	4.09±1.3	27.24±6.6	12.76±4.7	26.77±5.2	**22.0±2.5**
Standard NNs [17]		28.13±4.8	21.23±5.0	4.44±1.6	27.07±3.3	18.95±3.4	34.19±8.7	28.4±1.8
P-RBF NNS [17]	O=1	21.9±2.9	14.7 ± 2.1	2.9± 1.2	22.2±4.5	10.9±2.8	17.9±3.1	22.7±1.4
	O=2	24±2.1	14.8± 3.2	3.2± 1.2	35.2±5.8	12.2±3.1	17.4±5.3	23.3±1.2
PCA [17]		29.87 ±2.6	38.15±3.5	8.07±2.0	31.38±5.2	10.19±1.7	16.77±3.6	33.97±2.5
SVM [17]		34.78±0.0	44.44±0.0	37.43±0.0	24.14±0.0	**5.81±2.6**	18.23±4.4	30.0±0.0
LDA[17]		34.43±1.8	32.84±10.1	3.27±1.0	24.14±2.2	20.19±4.2	24.52±3.9	30.87±2.6
C4.5 trees [17]		32.43±3.1	29.63±7.2	8.25±2.3	33.62±6.0	8.38±1.5	36.45±7.6	30.7±0.9
Decision Tree [17]		27.39±3.6	24.81±6.1	7.13±1.8	24.83±2.2	11.43±2.8	31.45±8.6	26.47±1.5

Table 3. Results of comparative analysis with the other classification methods based on the boundary analysis (The best results are shown in boldface)

Classifiers			Pima	WDBC	WPBC	Ionosphere	Sonar
Proposed model	LSE	O=1	78.27±5.12	**97.18±1.71**	83.38±7.51	91.09±4.99	81.73±9.96
		O=2	78.37±4.01	97.02±2.31	80.01±9.96	91.23±4.61	79.39±5.60
	WLSE (q=1)	O=1	**78.39±2.14**	97.03±2.03	82.08±8.69	91.17±6.36	87.52±8.01
		O=2	78.40±3.01	97.01±2.22	79.23±8.42	90.62±5.59	87.52±5.99
	WLSE (q=2)	O=1	78.13±5.32	97.01±0.84	**83.44±10.8**	**91.46±3.62**	89.42±5.12
		O=2	78.12±4.06	97.02±2.19	79.02±10.6	90.89±3.52	**90.42±7.25**
LBDA based Classifier [15]	LBDA+NN(non)		76.16±0.06	94.90±0.05	77.34±0.14	88.36±0.29	84.32±0.06
	LBDA+NN(all)		70.43±0.08	95.62±0.04	73.16±0.20	**91.38±0.07**	86.42±0.12
SMO (SVM)[22]			76.63	96.85	Not Available	Not Available	Not Available
Robust SVM[24]			77±3.0	Not Available	77±3.0	Not Available	**90.0±3.0**
PDFC [24]			77±2.0	Not Available	0.81±7.0	Not Available	**90.0±2.0**

LBDA - linear boundary discriminant analysis. LBDA+NN(non) uses only non-boundary patterns to train Nearest Neighbor classifier, while LBDA+NN(all) uses all patterns to train the same classifier.

Acknowledgements

This work was supported by the GRRC program of Gyeonggi province [GRRC Suwon 2014-B2, Center for U-city Security & Surveillance Technology] and by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2012R1A1B 3003568).

References

[1] S. Albrecht, et al., "Generalized radial basis function networks for classification and novelty detection: self-organization of optimal Bayesian decision," *Neural Networks*, vol. 13, pp. 1075-1093, 2000.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, Berlin, 2006.

[3] B. Bouchon-Meunier, "Linguistic hedges and fuzzy logic", Fuzzy Syste*ms, 1992.*, *IEEE International Conference,* pp. 247-254, 1992.

[4] G. Bugmann, "Normalized Gaussian radial basis function networks," *Neurocomputing*, vol. 20, pp. 97-110, 1998.

[5] N. M. Duy and T.T. Cong, "Numerical solution of differential equations using multiquadric radial basis function networks," *Neural Networks*, vol. 14, pp. 185-199, 2001.

[6] M.J. Er, S.Q. Wu, J.W. Lu, and H.L. Toh, "Face recognition with radical basis function (RBF) neural networks," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 697-710, 2002.

[7] X. Hong, S. Chen, and C.J. Harris, "A kernel-based two-class classifier for imbalanced data sets," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 28-41, 2007.

[8] X.Y. Jing, Y.F. Yao, D. Zhang, J.Y. Yang, and M. Li, "Face and palm print pixel level fusion and Kernel DCV-RBF classifier for small sample biometric recognition," *Pattern Recognition*, vol. 40, pp. 3209-3224, 2007.

[9] N.B. Karayiannis, "Reformulated radial basis neural networks trained by gradient descent," *IEEE Transactions on Neural Networks,*vol. 10, no. 3, pp. 657-671, 1999.

[10] R. Krishapuram and J.M. Keller, "A possibilistic approach to clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 98-110, 1993.

[11] S.M. Loone and G. Irwin, "Improving neural network training solutions using regularization," *Neurocomputing*, vol. 37, pp. 71-90, 2001.

[12] C.G. Looney, "Radial basis functional link nets and fuzzy reasoning," *Neurocomputing*, vol. 48, pp. 489-509, 2002.

[13] L. Ma, A. Wahab, G.S. Ng, and S. Erdogan, "An experimental study of the extended NRBF regression model and its enhancement for classification problem," *Neurocomputing*, vol. 72, pp. 458-470, 2008.

[14] M. Lazaro, I. Santamaria, and C. Pantaleon, "A new EM-based training algorithm for RBF networks," *Neural Networks*, vol. 16, no. 1, pp. 69-77, 2003.

[15] Jin Hee Na, Myoung Soo Park, and Jin Young Choi, "Linear boundary discriminant analysis," *Pattern Recognition*, vol. 43, pp. 929-936, 2010.

[16] W.W.Y. Ng, A. Dorado, D.S. Yeung, W. Pedrycz, and E. Izquierdo, "Image classification with the use of radial basis function neural networks and the minimization of the localized generalization error," *Pattern Recognition*, vol. 40, pp. 19-32, 2007.

[17] B. J. Park, W. Pedrycz, and S.K. Oh, "Polynomial-based radial basis function neural networks (P-RBFNNs) and their application to pattern classify-cation," *Applied Intelligence*, vol. 32, No. 1, pp. 27-46, 2008.

[18] W. Pedrycz, "Conditional fuzzy clustering in the design of radial basis function neural networks," *IEEE Transaction on Neural Netw*orks, vol. 9 no. 4, pp. 601-612, 1998.

[19] W. Pedrycz, H.S. Park, and S.K. Oh, "A granular-oriented development of functional radial basis function neural networks," *Neurocomputing*, vol. 72, pp. 420-435, 2008.

[20] M.J.D. Powell, *Radial basis functions for multivariable interpolation: a review*, in: J.C. Mason, M.G. Cox (Eds.), Algorithms for Approximation, Oxford University Press, Oxford, pp. 143-167, 1987.

[21] M.R. Senapati, I. Vijaya, and P.K. Dash, "Rule Extraction from Radial Basis Functional Neural Networks by Using Particle Swarm Optimization," *Journal of Computer Science*, vol. 3 no. 8, pp. 592-599, 2007.

[22] A. Cervantes, I.M. Galvan, and P. Isasi, "AMSPO: A New Particle Swarm Method for Nearest Neighborhood Classification," IEEE Transaction on Systems, Man, and Cybernetics part B, vol. 39, No. 5, pp. 1082-1091, 2009.

[23] B. Bouchon-Meunier, "Linguistic hedges and fuzzy logic", *Fuzzy Systems, 1992., IEEE International Conference*, pp, 247-254, 1992.

[24] Y. Forghani, H. S. Yazdi, "Robust support vector machine-trained fuzzy system," Neural Networks, vol. 50, pp. 154-165, 2014.

[25] Hichem Frigui, Quiem Bchir, and Naouel Baili, "An Overview of Unsupervised and Semi-Supervised Fuzzy Kernel Clustering", *IJFIS*, vol. 13, no. 4, pp. 254-268, 2013.

Seok-Beom Roh received the B.Sc., M.Sc., and Ph.D. degrees in control and instrumentation engineering from Wonkwang University, Korea, in 1994, 1996, and 2006 respectively. He is currently a Senior manager with Wia Corporation Co. His research interests include fuzzy set, neural networks, genetic algorithms, computational intelligence and statistical learning.

Sung-Kwun Oh received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Yonsei University, Seoul, Korea, in 1981, 1983, and 1993, respectively. During 1983-1989, he was a Senior Researcher of R&D Lab. of Lucky-Goldstar Industrial Systems Co., Ltd. From 1996 to 1997, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada. He is currently a Professor with the Department of Electrical Engineering, University of Suwon, Suwon, South Korea. His research interests include fuzzy system, fuzzy-neural networks, automation systems, advanced computational intelligence, and intelligent control. He currently serves as an Associate Editor of the *KIEE Transactions on Systems and Control*, *International Journal of Fuzzy Logic* and *Intelligent Systems of the KFIS*, and *Information Sciences*.