Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 161 (2010) 1803–1822

FUZZY sets and systems

www.elsevier.com/locate/fss

The design methodology of radial basis function neural networks based on fuzzy K-nearest neighbors approach

Seok-Beom Roh[a], Tae-Chon Ahn[a], Witold Pedrycz[b,c,*]

[a]*Department of Electronic and Control Engineering, Wonkwang University, 344-2, Shinyong-Dong, Iksan, Chon-Buk 570-749, South Korea*
[b]*Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, Canada T6G 2G7*
[c]*Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland*

Received 16 August 2008; received in revised form 13 September 2009; accepted 22 October 2009
Available online 10 November 2009

Abstract

Various approaches to partitioning of high-dimensional input space have been studied with the intent of developing homogeneous clusters formed over input and output spaces of variables encountered in system modeling. In this study, we propose a new design methodology of a fuzzy model where the input space is partitioned by making use of some classification algorithm, especially, fuzzy K-nearest neighbors (K-NN) classifier being guided by some auxiliary information granules formed in the output space. This classifier being regarded in the context of this design as a supervision mechanism is used to capture the distribution of data over the output space. This type of supervision is beneficial when developing the structure in the input space. It is demonstrated that data involved in a partition constructed by the fuzzy K-NN method realized in the input space show a high level of homogeneity with regard to the data present in the output space. This enhances the performance of the fuzzy rule-based model whose premises in the rules involve partitions formed by the fuzzy K-NN. The design is illustrated with the aid of numeric examples that also provide a detailed insight into the performance of the fuzzy models and quantify several crucial design issues.

Keywords: Classification; Fuzzy K-NN; Auxiliary information granules; Supervised clustering; Cluster homogeneity

1. Introduction

Radial basis function (RBF) neural networks have attracted a lot of attention due to their learning and generalization abilities [26]. In the literature, we can encounter a vast body of knowledge on the analysis and design of RBF neural networks as well as their applications, cf. [7,16,23,36]. RBF neural networks form a suite of unified links between different research pursuits such as function approximation, regularization, noisy interpolation, density estimation, optimal classification theory, and potential functions [1].

In their design, RBF neural networks utilize overlapping localized regions formed in the input space and articulated by receptive fields (i.e., radial basis functions) to create complex and highly nonlinear decision boundaries. RBF neural networks encapsulate some domain knowledge that is captured by receptive fields [26]. The comparison of RBF neural

* Corresponding author at: Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, Canada T6G 2G7. Tel.: +1 7804923332; fax: +1 7804921811.

E-mail addresses: nado@wonkwang.ac.kr (S.-B. Roh), tcahn@wonkwang.ac.kr (T.-C. Ahn), pedrycz@ee.ualberta.ca (W. Pedrycz).

0165-0114/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2009.10.014

networks with fuzzy systems (and rule-based systems, in particular) with regard to the structure and the underlying learning algorithm has revealed some interesting analogies [10]. As there has been some research to elicit rules from the original data set when dealing with fuzzy models, some studies have been focused on building these receptive fields so that they significantly enhance the quality of the network as well as come with well-defined semantics.

Various models exist that describe design methods to construct fuzzy rule-based structures including heuristic, statistical, and adaptive techniques, see [13,15,17,22,27,33,35]. Some studies in this area have focused on problems of rule extraction by using neural networks, genetic algorithms, and a variety of clustering-oriented techniques [2,6,11,18–21,24,28,32,34].

Fuzzy C-means (FCM) has been quite often exploited to localize fuzzy (linguistic) terms and subsequently help construct fuzzy rules in the input space. Pedrycz [26] has pointed at a certain drawback of the original objective function-based clustering techniques in the context of system modeling. This shortcoming, which is commonly encountered when using the clustering methods based on the minimization of the objective function to develop linguistic terms of a fuzzy model in the input space, is that all of these terms are formed in a completely unsupervised manner even though there could be some available component of supervision associated with the dependent (output) variables. In other words, the main thrust of any unsupervised clustering techniques is somewhat different from the main requirement one has to cope with when constructing RBF neural networks.

To overcome this drawback, a method of *conditional* (context-based) fuzzy clustering where the dependent variables can be effectively exploited to develop meaningful clusters in the input space was proposed. While still forming clusters in the input space, the method is supervised by the existing information granules present in the output space and the ensuing clustering algorithm is guided by the presence of this structure. Staiano et al. [31] proposed a supervised fuzzy clustering (SFC) method, which formed another advancement of the generic form of the FCM again with the primary intent to enhance the modeling performance of RBF neural networks.

For these two clustering techniques such as the conditional fuzzy clustering and the supervised fuzzy clustering, the prototypes (centers of the clusters) are determined under a supervision mechanism conveyed by the output variable. Generally, there are two phases in the learning of RBF neural networks: a construction of receptive fields and the ensuing parametric learning.

The objective of this study is to thoroughly discuss the preprocessing phase of the design of RBF neural networks as it is done through the supervised version of the fuzzy K-NN and elaborate on its role in the enhancements of the performance of the network. It is worth noting that the exploitation of the dependent variable to develop the meaningful clusters in the input space (i.e., the projection of the auxiliary information on the input space to determine the clusters in the input space) can enhance the performance of the RBF neural networks.

In this study, we choose the fuzzy K-NN to project the auxiliary information available in the output space into the input space and use it effectively to form the receptive fields in the input space.

When dealing with new data patterns, in contrast to the clustering techniques which only depend on the prototypes without any auxiliary information, the K-NN approach is navigated by the information conveyed by the classification outcomes available through the nearest neighbors.

The fuzzy K-NN approach, as originally proposed by Keller, Gray, and Givens [3,4,12,30], was introduced as a classification algorithm realized at the level of the premise parts of the rules. The supervision effect comes in the form of information granules formed by clustering realized in unsupervised mode, such as K-means, FCM, or fuzzy C-regression clustering [14] being completed in the output space. The information granules satisfying the homogeneity requirement in the input space are extracted by using the classification algorithm. In the sequel, the proposed model is experimentally contrasted with several representative categories of neurofuzzy models already known in the literature.

The paper is arranged as follows. First, in Section 2, we introduce RBF neural networks which take advantage of the use of the fuzzy K-NN supported by auxiliary information. The issue of homogeneity of fuzzy K-NN is elaborated in Section 3. Section 4 presents experimental results. Conclusions are covered in Section 5.

2. The design of RBF neural networks based on fuzzy K-nearest neighbors

It is well known that the clustering is often used as a preprocessing phase in the design of RBF neural networks. In general, clustering is predominantly focused on capturing and describing the distribution of data (patterns) being located in the space of input variables (input space). Cluster formation that minimizes some predefined objective function is accomplished without any supervision (i.e., without resorting to the use of the dependent variables). In this sense, unsupervised learning ignores any useful information which otherwise could have been used effectively in the clustering process [4].

When dealing with the design of RBF neural networks, it becomes beneficial to take into account information about continuous labeling (output) of patterns so that one could come up with clusters which are meaningful from the standpoint of the realization of the nonlinear mapping formed by the constructed neural network.

To elaborate on the essence of the method, let us consider a set of patterns $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, $\mathbf{x}_k \in \Re^m$ (where m stands for the dimensionality of the input space) along with an auxiliary data set $Y = \{y_1, y_2, \ldots, y_N\}$, $y_k \in \Re$. Each element of X is associated with the element in the set Y (i.e., $\mathbf{x}_k$ is related to y_k).

2.1. Fuzzy K-nearest neighbors with auxiliary information for preprocessing in RBF neural networks

Let us consider the activation levels of data (patterns) generated by the already studied clustering techniques such as the conditional fuzzy C-means (CFCM) clustering [25] and the supervised fuzzy clustering (SFC) [31].

The iterative optimization scheme used in the CFCM is composed of the two update formulas using which we successively modify the partition matrix and the prototypes

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

here, $f_k^l = B_l(y_k)$:

$$\mathbf{v}_i^l = \frac{\sum_{k=1}^{N} (u_{ik}^l)^p \cdot \mathbf{x}_k}{\sum_{k=1}^{N} (u_{ik}^l)^p} \tag{2}$$

where u_{ik}^l and $\mathbf{v}_i^l$ are the element of the partition matrix and the ith cluster associated with the lth auxiliary information respectively, r is the number of rules (clusters) formed for this auxiliary information, N is the number of data patterns, C denotes the number of auxiliary information granules, and p means the fuzzification coefficient. Finally, B_l is the linguistic term, which defines the lth auxiliary information (information granule).

For SFC, the partition matrix and the prototypes are updated iteratively in the following fashion:

$$u_{ik} = \frac{1}{\sum_{j=1}^{C} \left(\dfrac{\|\mathbf{x}_k - \mathbf{v}_i\|^2 + \alpha(\hat{y}_k - y_k)^2}{\|\mathbf{x}_k - \mathbf{v}_j\|^2 + \alpha(\hat{y}_k - y_k)^2} \right)^{1/(p-1)}}, \quad 1 \le i \le C, \ \ 1 \le k \le N \tag{3}$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^p \cdot \mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^p} \tag{4}$$

The parameter α is a weight factor (weight), which is used to control the impact of the structure determined in the output space during the clustering process carried out in the input space.

Assuming that the linguistic term B is defined in the output space and considering that the training data set is given, these activation levels could be determined.

For the testing data, as the value of the dependent variable is not known, (1) and (3) have to be modified to accommodate this fact. Here the calculations are carried out as follows:

$$u_{ik} = \frac{1}{\sum_{j=1}^{C} \left(\dfrac{\|\mathbf{x}_k - \mathbf{v}_i\|}{\|\mathbf{x}_k - \mathbf{v}_j\|} \right)^{2/(p-1)}} \tag{5}$$

The merit of the conditional fuzzy clustering and the supervised fuzzy clustering stems from the fact that these clustering approaches can reflect upon the conditional information residing within the output variable when constructing the clusters in the input space.

To fully take advantage of the conditional information (auxiliary information or the distribution of clusters in the output space) when realizing the assignment of clusters in the input space, we use the fuzzy K-nearest approach with auxiliary information.

2.1.1. Auxiliary information in the output space

To supervise the computing realized by the fuzzy K-NN, we consider three mechanisms of information granulation such as K-means, FCM, and the fuzzy C-regression model.

First, let us consider K-means and FCM for information granulation realized in the output space.

When running K-means and its fuzzy counterpart, we can construct an apex of each auxiliary information granule. After that, we calculate the activation levels of each auxiliary information granule based on these extracted apexes

$$f_j^i = g_i(y_j, \mathbf{v}), \quad 1 \leq i \leq C, \ 1 \leq j \leq N \tag{6}$$

where $\mathbf{v} = [v_1 \ v_2 \ \cdots \ v_C]$, $v_j \in \Re$ is a vector of the apexes of the *i*th information granule (cluster) which is defined on the output space while f_j^i stands for the activation level of the *i*th information granule related to the *j*th data y_j.

The term on the right-hand side of (6), $g_i(y_j, \mathbf{v})$, is determined based on the predefined shape of the information granules (say, triangular or bell-like shape).

When we use the triangular information granules, $g_i(y_j, \mathbf{v})$ is defined as

$$g_i(y_j, \mathbf{v}) = \begin{cases} 0 & \text{if } y_j < v_{i-1} \text{ or } y_j > v_{i+1} \\ \dfrac{y_j - v_{i-1}}{v_i - v_{i-1}} & \text{if } v_{i-1} \leq y_j < v_i \\ \dfrac{y_j - v_{i+1}}{v_i - v_{i+1}} & \text{if } v_i \leq y_j < v_{i+1} \end{cases} \tag{7}$$

For the bell-like shape information granules, $g_i(y_j, \mathbf{V})$ is expressed as

$$g_i(y_j, \mathbf{v}) = \frac{1}{\sum_{l=1}^{C} \left(\dfrac{\|y_j - v_i\|}{\|y_j - v_l\|} \right)^{2/(p-1)}} \tag{8}$$

Now, let us consider fuzzy C-regression clustering. While K-means and FCM develop hyper-spherical shapes of clusters, fuzzy C-regression clustering supports the hyper-planar geometry of clusters. Let us consider a set of N data points, $(\mathbf{x}_k, y_k)$, $1 \leq k \leq N$ where $\mathbf{x}_k = [x_{1k} \ x_{2k} \ \cdots \ x_{mk}] \in \Re^m$.

We assume C hyper-plane clusters which are represented in the following form:

$$\hat{y}_k^1 = \mathbf{w}_k \mathbf{a}^1, \ \hat{y}_k^2 = \mathbf{w}_k \mathbf{a}^2, \ldots, \hat{y}_k^C = \mathbf{w}_k \mathbf{a}^C \tag{9}$$

where $\mathbf{w}_k = [1 \ \mathbf{x}_k] \in \Re^{m+1}$.

The objective function used in the fuzzy C-regression clustering has been modified in comparison with the one used in the “standard" FCM and reads now as follows:

$$J = \sum_{i=1}^{C} \sum_{k=1}^{N} (u_{ik})^p \cdot \|y_k - \hat{y}_k^i\|^2 = \sum_{i=1}^{C} \sum_{k=1}^{N} (u_{ik})^p \cdot \|y_k - \mathbf{w}_k \mathbf{a}_i\|^2 \tag{10}$$

$$\sum_{i=1}^{C} u_{ik} = 1 \tag{11}$$

The constraint imposed on the partition matrix used in the fuzzy C-regression clustering is expressed by (11). To arrive at the underlying structure by minimizing (10), we need to estimate the parameter vector $\mathbf{a}_i$ and the components of the partition matrix u_{ik}. The membership degrees u_{ik} are calculated as

$$u_{ik} = \frac{1}{\sum_{j=1}^{C} \left(\dfrac{(y_k - \hat{y}_k^i)^2}{(y_k - \hat{y}_k^j)^2} \right)^{1/(p-1)}} = \frac{1}{\sum_{j=1}^{C} \left(\dfrac{(y_k - \mathbf{w}_k \mathbf{a}_i)^2}{(y_k - \mathbf{w}_k \mathbf{a}_j)^2} \right)^{1/(p-1)}}, \quad 1 \leq i \leq C, \ 1 \leq k \leq N \tag{12}$$

where the parameter vector $\mathbf{a}_i$ is estimated by applying the weighted least square method thus leading to the well-known expression

$$\mathbf{a}_i = (\mathbf{W}^T\mathbf{U}_i\mathbf{W})^{-1}(\mathbf{X}^T\mathbf{U}_i\mathbf{Y}) \tag{13}$$

where

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_N \end{bmatrix} \in \Re^{N\times(m+1)}, \quad \mathbf{U}_i = \begin{bmatrix} u_{i1} & 0 & \cdots & 0 \\ 0 & u_{i2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & u_{iN} \end{bmatrix} \in \Re^{N\times N} \quad \text{and} \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \Re^{N\times 1}$$

2.1.2. Fuzzy K-nearest neighbors approach

After forming auxiliary (conditional) information granules in the output space, we have to reveal structure (clusters) in the input space by taking into account the granular findings obtained so far (i.e., f_j^i). To proceed with this task, we consider using fuzzy K-NN approach. Usually, the K-NN classification classifies any unknown sample by comparing it to its nearest neighbors among a set of labeled samples (training data).

Fuzzy K-NN replaces the generic K-NN classification rule by associating each sample with a membership value expressing how closely the pattern belongs to a given class. The corresponding degrees of membership are given in the form

$$\mu_i(\mathbf{x}) = \frac{\sum_{j=1}^{K} f_j^i \left(\dfrac{1}{\|\mathbf{x}-\mathbf{x}_j^*\|^{2/(p-1)}} \right)}{\sum_{l=1}^{K} \left(\dfrac{1}{\|\mathbf{x}-\mathbf{x}_l^*\|^{2/(p-1)}} \right)} = \sum_{j=1}^{K} f_j^i \cdot \left(\frac{1}{\sum_{l=1}^{K} \left(\dfrac{\|\mathbf{x}-\mathbf{x}_j^*\|}{\|\mathbf{x}-\mathbf{x}_l^*\|} \right)^{2/(p-1)}} \right) = \sum_{j=1}^{K} f_j^i \cdot D_j(\mathbf{x}) \tag{14}$$

$$D_j(\mathbf{x}) = \frac{1}{\sum_{l=1}^{K} \left(\dfrac{\|\mathbf{x}-\mathbf{x}_j^*\|}{\|\mathbf{x}-\mathbf{x}_l^*\|} \right)^{2/(p-1)}} \tag{15}$$

where $\mathbf{x}$ is the pattern to be classified and $\mathbf{x}_j^*$ is the jth neighbor of the given data pattern $\mathbf{x}$. When a new data pattern is provided, we choose the K nearest neighbors $(\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots \mathbf{x}_K^*)$ of the given data pattern among the samples over the training data set from the viewpoint of the distance between the samples and the newly given data pattern.

In (14), the distance $\|\cdot\|$ can be treated as the Euclidean distance or its generalized weighted version.

If we assume that $\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_k^*$ are the prototypes of clusters and the unknown data pattern $\mathbf{x}$ is given, (15) is equivalent to (5) using which we compute the activation levels of clusters implied by the data $\mathbf{x}$.

Fuzzy K-NN generalizes the binary class allocation supported by the traditional K-NN method and by doing this often produces more accurate results. However, there is still a certain drawback that the value of the fuzzification coefficient p being used to scale the effect of the distance between unknown $\mathbf{x}$ and known $\mathbf{x}_j^*$ is selected arbitrarily.

From the design standpoint, there are several essential parameters of the fuzzy K-NN that impacts its usage of the produced results. These parameters concern the number of clusters (C), the values of the fuzzification coefficient (p) and the number of nearest neighbors (K) used in the procedure. The fuzzification coefficient exhibits a significant impact on the form (shape) of the developed clusters. The commonly used value of "p" is equal to 2. Lower values of the fuzzification coefficient produce more Boolean-like shapes of the fuzzy sets where the regions of intermediate membership values are very much reduced. When we increase the values of "p" above 2, the resulting membership functions start to become "spiky" with the values close to 1 in a very close vicinity of the prototypes. The number of nearest neighbors has an effect on the estimation ability of the fuzzy K-NN approach. In the extreme case where the number of neighbors is equal to the number of data patterns, it is sure that there is redundant calculation owing to some data patterns far from the query point. These data points far from the query point have little effect on the estimation result. When the number of nearest neighbors decreases to 1, the estimation result of fuzzy K-NN for the testing data becomes worse [8]. In addition, the number of clusters (C) also has an effect on the modeling performance of RBF

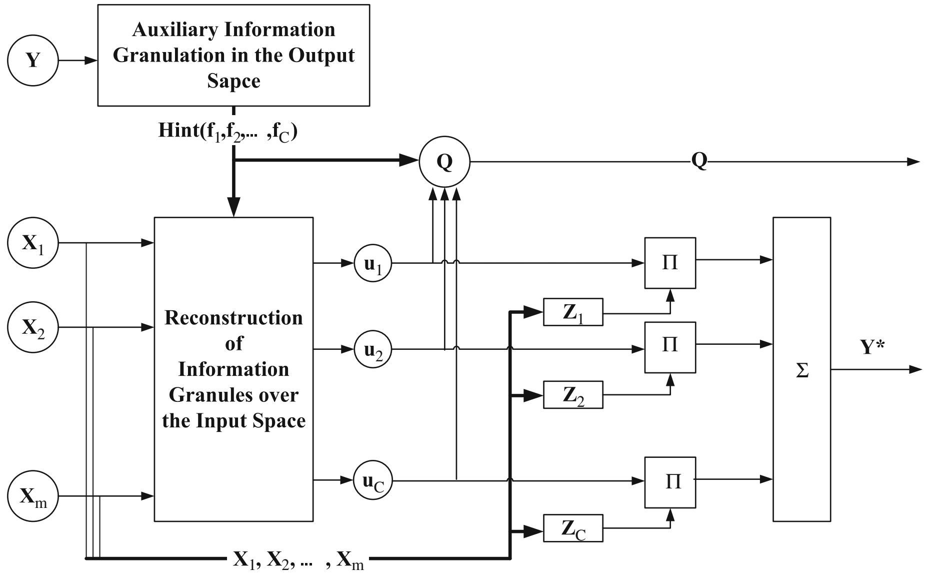

Fig. 1. Overall topology of a fuzzy model based on the fuzzy K-NN.

networks. Generally, it is well known that the larger the number of clusters, the worse the generalization abilities of the produced networks.

The choice of suitable numeric values of "p", "K", and "C" can be made on a basis of experimental studies.

2.2. RBF neural networks based on the fuzzy K-nearest neighbors approach

Now we elaborate on a new design methodology of RBF neural networks which relies on the use of the fuzzy K-NN being treated here as an integral design approach. As mentioned before, if we can reconstruct information granules over the input space without loss of homogeneity of the auxiliary information granules (clusters) constructed in the output space, the reconstructed information granules over the input space can help enhance the performance of the fuzzy model [26]. The auxiliary information granules are used as the guidance for reconstructing the information granules in the input space when running the fuzzy K-means. Fig. 1 depicts the overall topology of a fuzzy model involving the use of the fuzzy K-NN.

In Fig. 1, the circle labeled by "Y" denotes the output variable, u_i is the activation level of the ith cluster, the Hint($\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_C$) are the activation levels of the auxiliary information granules. $Z_1, Z_2, \ldots, Z_C$ denote the local models represented by the constant or the linear function

$$\text{constant}: \quad Z_i(\mathbf{x}) = a_i \tag{16a}$$

$$\text{linear function}: \quad Z_i(\mathbf{x}) = a_{i0} + \sum_{k=1}^{m} a_{ik} x_k \tag{16b}$$

In Fig. 1, $\prod$ and $\sum$ are used to denote the multiplication operator and summation, respectively.

Here Y^* denotes the output of the fuzzy model. The performance index Q, or its scaled (normalized) version quantifying the performance of the model has been introduced in the literature [28] and is governed by the expression

$$Q = \sum_{k=1}^{n} (\mathbf{u}_k - \mathbf{f}_k)^T (\mathbf{u}_k - \mathbf{f}_k) \tag{17a}$$

$$\tilde{Q} = \sum_{k=1}^{n} (\mathbf{u}_k - \mathbf{f}_k)^T (\mathbf{u}_k - \mathbf{f}_k)/N \tag{17b}$$

$$\mathbf{u}_k = [u_{1k} \;\; u_{2k} \;\; \cdots \;\; u_{Ck}]^T, \quad \mathbf{f}_k = [f_k^1 \;\; f_k^2 \;\; \cdots \;\; f_k^C]^T$$

The local model Z_i is a constant or a linear regression model. The output of the fuzzy model aggregates the results obtained at the level of the local models

$$y_k^* = \frac{\sum_{i=1}^{C} u_{ik} \cdot Z_i(\mathbf{x}_k)}{\sum_{i=1}^{C} u_{ik}} = \sum_{i=1}^{C} u_{ik} \cdot Z_i(\mathbf{x}_k) \tag{18}$$

where u_{ik} is the activation level of a fuzzy rule (i.e., information granule over the input space) related to input $\mathbf{x}_k$ and can be calculated using (14) and (15). Parameter estimation of the local models Z_i, $1 \le i \le C$ is realized by running the standard LSE method.

The detailed algorithm supporting the design of the proposed fuzzy model with fuzzy rule extraction by using fuzzy K-NN can be outlined as follows.

(1) Determine the parameters of the fuzzy K-NN classification, namely
  (a) The number of nearest neighbors (*K*).
  (b) The fuzzification coefficient (*p*).
(2) Determine the structure of the fuzzy model, that is:
  (a) The number of rules (*C*) which is the same as the number of the auxiliary information granules formed over the output space.
  (b) The type of the consequent polynomial (in this study, we use constant or linear function as a consequent part).
(3) Form the auxiliary information granules over the output space using one of the existing techniques, say
  (a) K-means.
  (b) Fuzzy C-means.
  (c) Fuzzy C-regression (FCRM) clustering.
(4) Reconstruct the information granules obtained at step 3 in the input space by using fuzzy K-NN approach:
  (a) We construct the information granules (i.e., estimate the activation level of each fuzzy rule) over the input space based on the auxiliary information granules (i.e., hint) pre-obtained over the output space by fuzzy K-NN approach.
(5) Estimate the values of the coefficients of the consequent polynomial of fuzzy rules to minimize the objective function by using the least square error (LSE) estimation technique.

3. Homogeneity with respect to clusters obtained through K-nearest neighbors approach

Generally all elements involved in each cluster are known to be homogeneous with respect to the prototype of the cluster. In other words, all elements involved in a cluster are the closest to the prototype of the cluster and distant from the prototypes of other clusters. When we deal with unlabeled data, the homogeneity of clusters can be implied by using the "conventional" unsupervised clustering [26]. If we consider the homogeneity of a labeled data set, we cannot assure that the clusters obtained by using unsupervised clustering methods are still homogeneous with respect to the labels of the patterns themselves. Especially, when we consider a regression task and the data sets *X* and *Y* (i.e., the same as the data sets in Section 2.1) are given, the homogeneity of a cluster in the sense of the dependent variable (the continuous label of the data pattern) is important to the modeling abilities and the effectiveness of the learning process [26]. Generally, the division of the input space is realized through fuzzy partitioning being the result of fuzzy clustering. Our concern is to come up with simple local models and this is possible if there is not too much variability of the output when we confine ourselves to the clustering being guided by the homogeneity in the output space. Given this, we are not concerned with the disconnected regions in the input space as this might be the specificity of the input–output mapping.

In this section, we compare the homogeneity from the viewpoint of the label of data of clustering approaches such as the existing clustering and the proposed fuzzy K-NN approach using a simple example. To illustrate the homogeneity of clustering approaches, let us consider a simple data set generated by the following relationship:

$$y = \sin(2\pi x) \tag{19}$$

where $x \in [0\ 1]$ and the values of the independent variable *x* are generated randomly.

First, we should construct conditional (auxiliary) information granules in the output space (here we use FCM). Fig. 2 shows the distribution of clusters which are represented by fuzzy sets B_1, B_2 and B_3 defined over the output

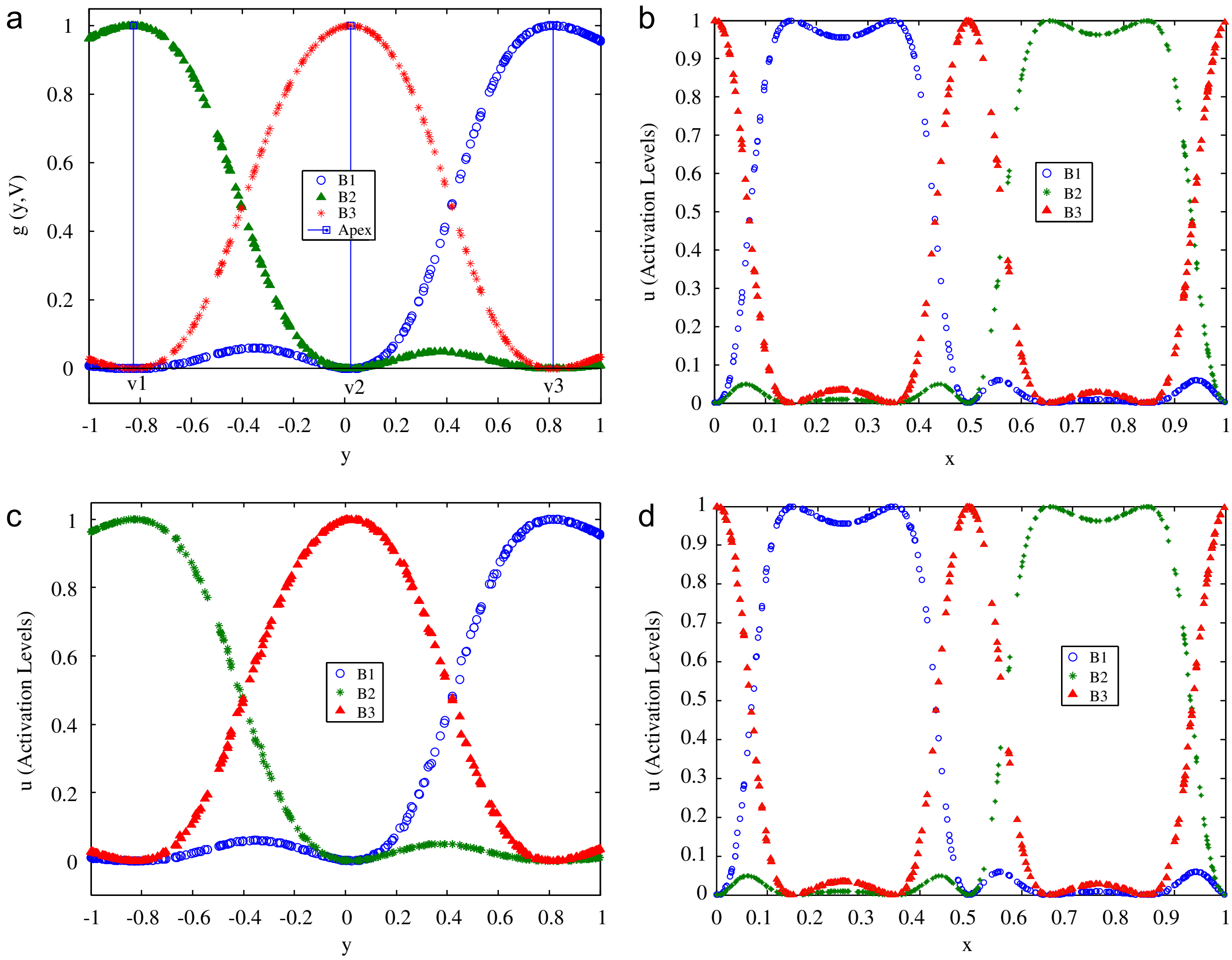

Fig. 2. Activation levels of clusters (i.e., auxiliary information) obtained by running FCM in the output space (y). (a) The auxiliary information versus output variable; (b) the auxiliary information versus input variable while adopting K-nearest neighbors in the input space; (c) versus output variable; (d) versus input variable.

space. Fig. 2(a) visualizes auxiliary information, which is extracted in the output space by using clustering (FCM) with regard to the output variable while Fig. 2(b) shows a distribution of the same auxiliary information with regard to the input variable.

Fig. 2(c) shows the activation levels of auxiliary information granules present in the output space while Fig. 2(d) displays the activation levels obtained by fuzzy K-NN in the input space with the fuzzification coefficient $p = 2$.

After forming the auxiliary information granules, we assign the clusters in the input space considering the effect of such auxiliary information granules.

We use the conditional fuzzy clustering to form some meaningful clusters in the input space under the consideration of the auxiliary information. Fig. 3 shows the apexes and activation levels of each cluster obtained by using conditional clustering with auxiliary information granules. The number of clusters for each auxiliary cluster is equal to 2 and the total number of clusters formed in the input space is equal to 6. When inspecting the homogeneity of clusters formed by the conditional fuzzy clustering with the respect to the labels (i.e., auxiliary information extracted by using clustering over the output space), we observe how well the clusters formed over the input space by the conditional fuzzy clustering are similar to the auxiliary information extracted in the output space (for comparison refer to the graph presented in Figs. 2 and 3).

We note that there is no similarity between the activation levels determined by the conditional fuzzy clustering in the input space and the auxiliary information defined in the output space. The clusters formed by the conditional fuzzy

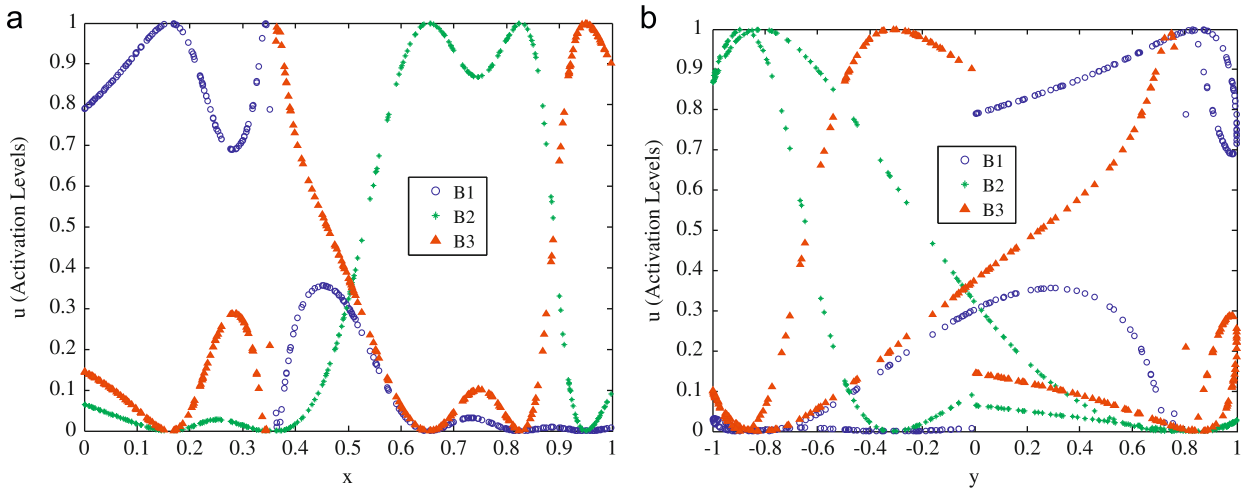

Fig. 3. The activation levels of each auxiliary information (activation level of auxiliary information calculated by summing up the activation levels of all clusters involved in auxiliary information, (a) activation level versus input variable, (b) activation level versus output variable).

Table 1
The values of the parameters of the proposed model.

Predefined parameters	Value
Number of auxiliary information granules (C)	2–12
Order of the polynomial (O)	0 (constant) or 1 (linear function)
Number of nearest neighbors (K)	2–7
Fuzzification coefficient (p)	1.2–4.0 with interval 0.2
Number of clusters for RBFNN_CFCM or fuzzy number of rules for RBFNN_FCM and RBFNN_SFC (R)	2–12

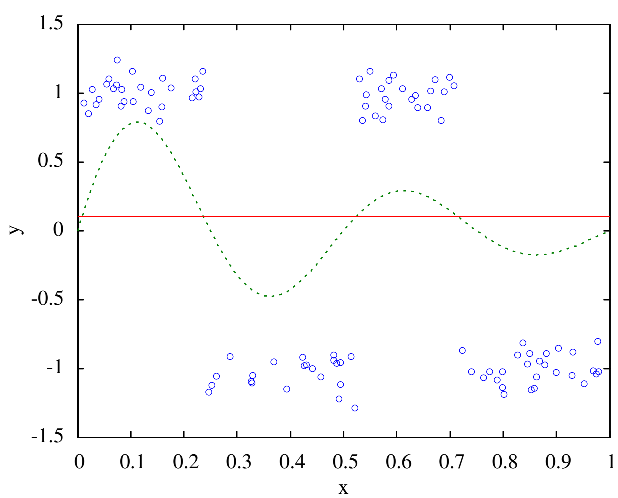

Fig. 4. Relationship between input and output variables for synthetic data.

clustering are not homogeneous with respect to the auxiliary information. To show the reconstruction performance of the fuzzy K-NN, we apply fuzzy K-NN method to the same data set and compare the activation levels determined by using fuzzy K-nearest neighbors with those coming from the application of the conditional fuzzy clustering.

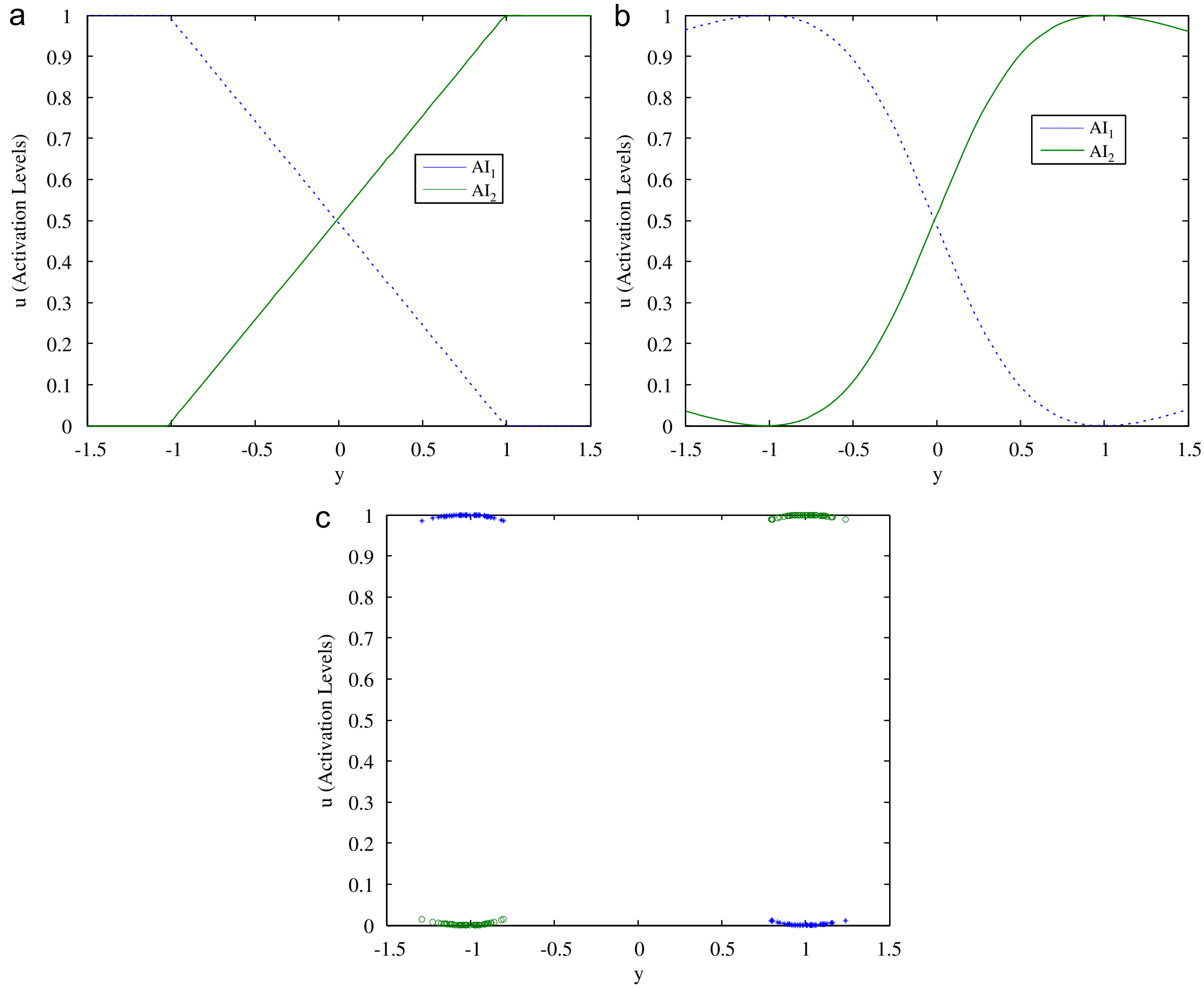

Fig. 5. Activation levels of the auxiliary information granules formed in the output space obtained in case of the use of different clustering techniques. (a) K-means clustering; (b) FCM clustering; (c) fuzzy C-regression clustering.

From Figs. 2 and 3, we can conclude that fuzzy K-NN leads to the formation of more homogeneous (similar) clusters than those produced by the conditional fuzzy C-means clustering. Also by inspecting the values of the performance index Q, see (17), we can note that the fuzzy K-NN ($Q = 0.339$) performs better than the conditional fuzzy C-means clustering (where $Q = 16.216$).

4. Experimental studies

In the series of experiments, we consider a fuzzy model with several receptive fields which are formed in the input space with the use of the fuzzy K-NN whose auxiliary information granules are realized by K-means, FCM or fuzzy C-regression clustering completed in the output space. These experiments are used to evaluate the effectiveness of the proposed model.

The first experiment deals with two synthetic data sets. In other experiments we use some selected data sets coming from the machine learning repository (http://www.ics.uci.edu/~mlearn/MLSummary.html). For machine learning data sets, we randomly split the entire data set into the training (60%) and testing part (40%) for the second and third data collection, respectively. We carry out an experiment with the predefined values of the parameters, see Table 1.

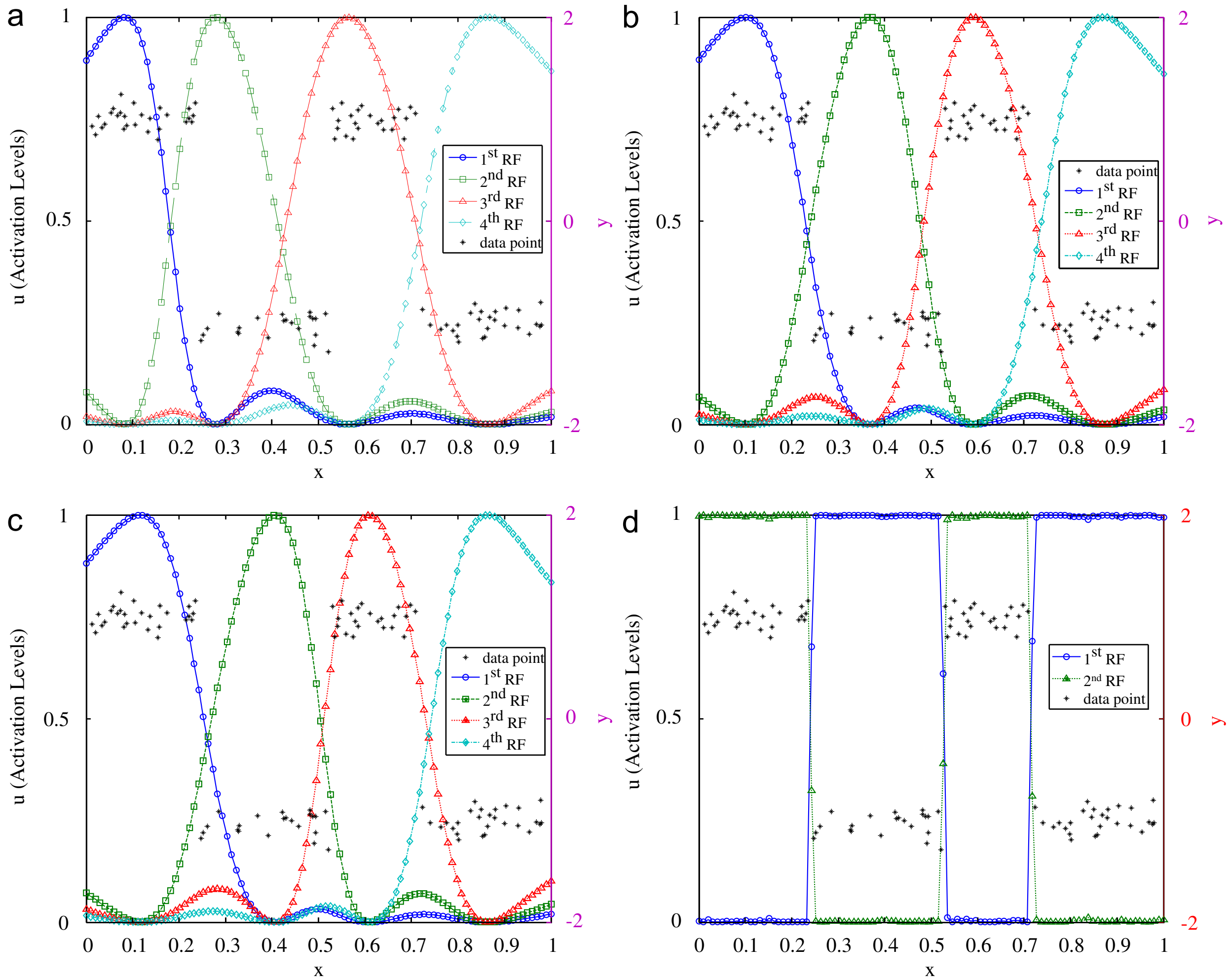

Fig. 6. Activation levels of clusters obtained by (a) fuzzy C-means clustering; (b) supervised fuzzy clustering; (c) conditional fuzzy clustering; (d) fuzzy K-NN.

The performance of the model are quantified in terms of the root mean square error (RMSE),

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2} \tag{20}$$

4.1. Synthetic one-dimensional data

We consider a collection of synthetic data governed by the following expression:

$$y = \begin{cases} 1 + N(0, 0.1) & \text{if } e^{-2x}\sin(4\pi x) > \sigma \\ -1 + N(0, 0.1) & \text{if } e^{-2x}\sin(4\pi x) \le \sigma \end{cases} \tag{21}$$

Here, the values of x are confined to $X = [0, 1]$ and $N(a, \Sigma)$ is the normal distribution with mean a and variance Σ. Each training and testing data set is composed of 100 input–output data pairs. These data sets were generated randomly from the universe of discourse X. The plot of this nonlinear function is shown in Fig. 4. The data in the input space are more or less uniformly distributed in the input spaces. However, considering the distribution of data in the sense of

Table 2
Results of comparative analysis.

Model	Auxiliary information	Structure and parameters of model	PI(RMSE)	EPI(RMSE)	Q (training data)	Q (testing data)
RBFNN _FCM	N/A	$O = 0$, $p = 1.6$, $R = 4$	0.482 ± 0.076	0.574 ± 0.051	N/A	N/A
	N/A	$O = 0$, $p = 1.8$, $R = 4$	0.399 ± 0.0424	0.524 ± 0.070	N/A	N/A
RBFNN_SFC	N/A	$O = 0$, $p = 1.4$, $R = 4$	0.498 ± 0.115	0.582 ± 0.080	N/A	N/A
	N/A	$O = 1$, $p = 1.8$, $R = 4$	0.392 ± 0.041	0.549 ± 0.083	N/A	N/A
RBFNN _CFCM	HCM	$O = 0$, $p = 2.0$, $C = 2$, $R = 2$	0.432 ± 0.045	0.508 ± 0.036	0.099 ± 0.019	0.129 ± 0.022
		$O = 1$, $p = 2.2$, $C = 2$, $R = 2$	0.413 ± 0.049	0.518 ± 0.063	0.105 ± 0.027	0.137 ± 0.033
	FCM	$O = 0$, $p = 2.0$, $C = 2$, $R = 2$	0.425 ± 0.047	0.527 ± 0.073	0.103 ± 0.026	0.148 ± 0.040
		$O = 1$, $p = 1.4$, $C = 2$, $R = 2$	0.381 ± 0.060	0.527 ± 0.080	0.095 ± 0.041	0.161 ± 0.041
	FCRM	$O = 0$, $p = 1.6$, $C = 2$, $R = 2$	0.411 ± 0.054	0.523 ± 0.084	0.094 ± 0.027	0.145 ± 0.051
		$O = 1$, $p = 1.8$, $C = 2$, $R = 2$	0.429 ± 0.027	0.527 ± 0.056	0.042 ± 0.032	0.044 ± 0.044
Proposed model (RBFNN_FKNN)	HCM	$O = 0$, $p = 3.2$, $C = 2$, $K = 5$	0.057 ± 0.006	0.410 ± 0.138	0 ± 0	0.082 ± 0.048
		$O = 1$, $p = 1.4$, $C = 2$, $K = 4$	0.058 ± 0.005	0.406 ± 0.085	0 ± 0	0.075 ± 0.031
	FCM	$O = 0$, $p = 3.2$, $C = 2$, $K = 7$	0.112 ± 0.012	0.417 ± 0.090	0 ± 0	0.071 ± 0.029
		$O = 1$, $p = 2.8$, $C = 2$, $K = 5$	0.100 ± 0.008	0.421 ± 0.068	0 ± 0	0.078 ± 0.027
	FCRM	$O = 0$, $p = 3.8$, $C = 2$, $K = 3$	0.143 ± 0.007	0.403 ± 0.104	0 ± 0	0.061 ± 0.030
		$O = 1$, $p = 4.0$, $C = 2$, $K = 3$	0.143 ± 0.0012	0.476 ± 0.050	0 ± 0	0.079 ± 0.019

the output variable, the data patterns are grouped into two distinct groups (around 1 and around −1). By considering this example, we demonstrate the effectiveness of the proposed method when forming the receptive fields of the RBF network.

In Fig. 4, the dotted line shows the function $e^{-2x}\sin(4\pi x)$ while the circles represent individual data (x_k, y_k). The line depicts the constant relationship $y = \sigma$, where σ is the threshold value defined in (21). Here, σ is equal to 0.1.

First, we construct a collection of auxiliary information granules in the output space by running various clustering methods. Fig. 5 depicts the activation levels (membership grades) produced by the K-means method generic FCM, and fuzzy C-regression clustering (with the fuzzification coefficient $p = 2$ and the number of clusters set to 2, $C = 2$). Those membership functions are shown vis-à-vis the dependent variable (y).

Next, we construct the receptive fields of the RBF. Here the auxiliary information is extracted by applying fuzzy C-means clustering over the output space. Fig. 6 shows the corresponding activation levels produced by the four methods such as (a) FCM, (b) supervised fuzzy clustering, (c) CFCM and (d) fuzzy K-NN. Figs. 6(a) and (b) display the activation levels generated by fuzzy clustering without the auxiliary information granules. Fig. 6(c) depicts the activation levels of receptive fields formed by the conditional fuzzy clustering. From Fig. 6(c), we observe that there are two auxiliary information granules and each auxiliary information gives rise to the two groups. Fig. 6(d) depicts the clusters formed by using fuzzy K-NN, which exploits the auxiliary information granules as shown in Fig. 5(b).

In Fig. 6, the asterisks denote the individual data and RF stands for the receptive field. Considering how well the groups are constructed (i.e., how much data patterns involved in a cluster are similar in the sense of the value of output variable (Y)), although the number of clusters formed by fuzzy K-NN is 2 (that is smaller than encountered in some other approaches), the clusters constructed by fuzzy K-NN approach shows better performance. Table 2 summarizes the performance of the proposed model vis-à-vis other models that are RBFNN with generic fuzzy C-means clustering, context based clustering (i.e., context fuzzy C-means clustering) or the supervised fuzzy clustering. Also the performance index Q is included that quantifies how well the information granules are reconstructed on the input space. Here PI and EPI are used to denote the values of the performance index for the training and testing set, respectively.

In Table 2, RBFNN_FCM, RBFNN_SFC, RBFNN_CFCM, and RBFNN_FKNN denote RBFNNs with receptive fields that are formed by running the generic FCM clustering, supervised fuzzy clustering, conditional fuzzy C-means

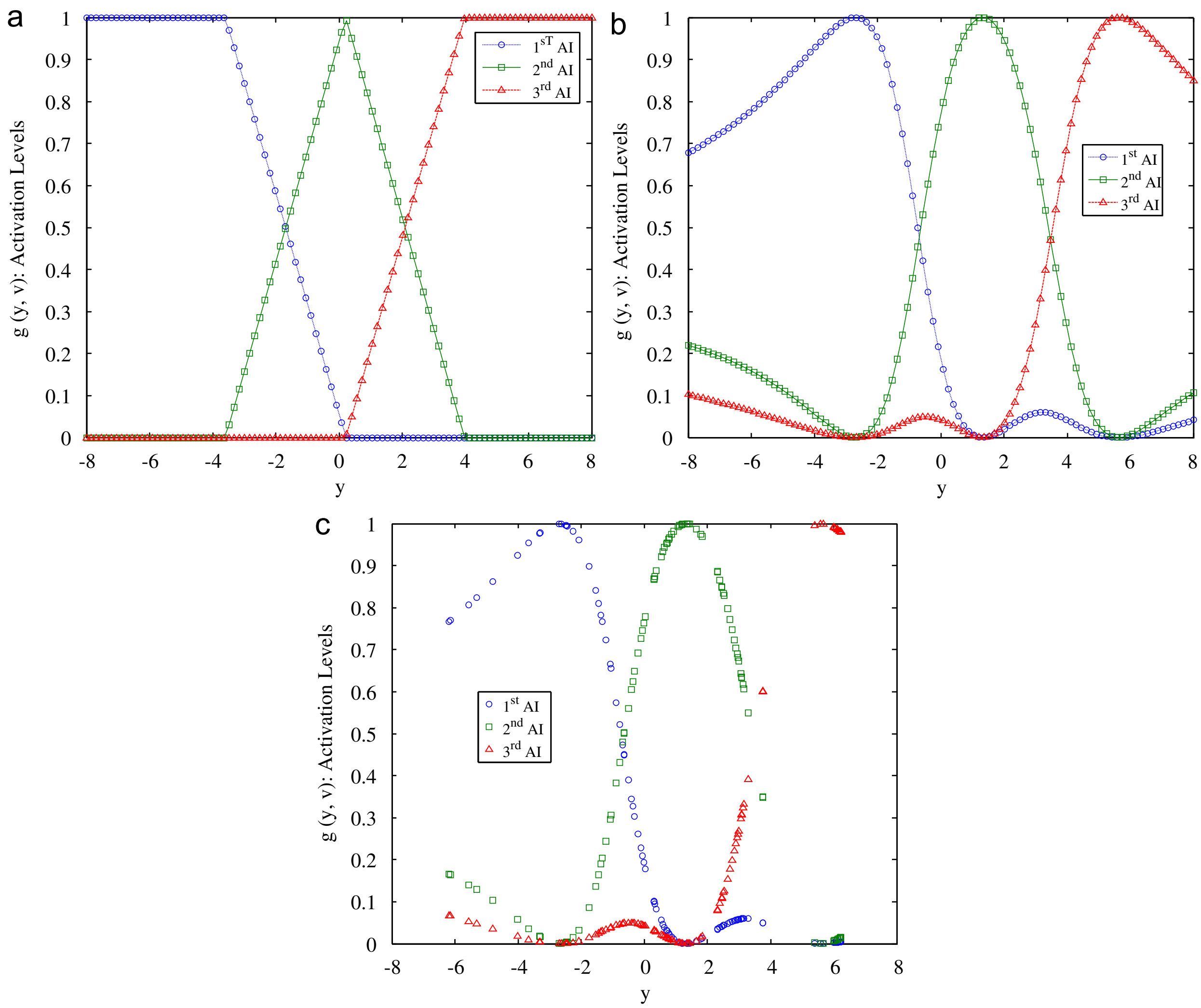

Fig. 7. Activation levels of the auxiliary information granules in the output space. (a) C-means clustering; (b) FCM clustering; (c) fuzzy C-regression clustering.

clustering, and fuzzy K-NN approach, respectively. The abbreviation HCM used in the table above stands for the K-means clustering.

4.2. Synthetic two-dimensional data

Let us consider a nonlinear function with two input variables described in the following way:

$$y = f(x_1, x_2) = \frac{(x_1 - 2)(2x_1 + 1)}{1 + x_1^2} \frac{(x_2 - 2)(2x_2 + 1)}{1 + x_2^2} \tag{22}$$

This function is defined over $[-5, 5] \times [-5, 5]$.

Subsequently Fig. 7 shows the auxiliary information granules formed in the output space when using K-means, generic FCM and the fuzzy C-regression clustering.

As before, the auxiliary information granules are essential when setting up some clusters in the input space. Fig. 8 visualizes the receptive fields of the RBFNN realized with the aid of the already discussed clustering techniques

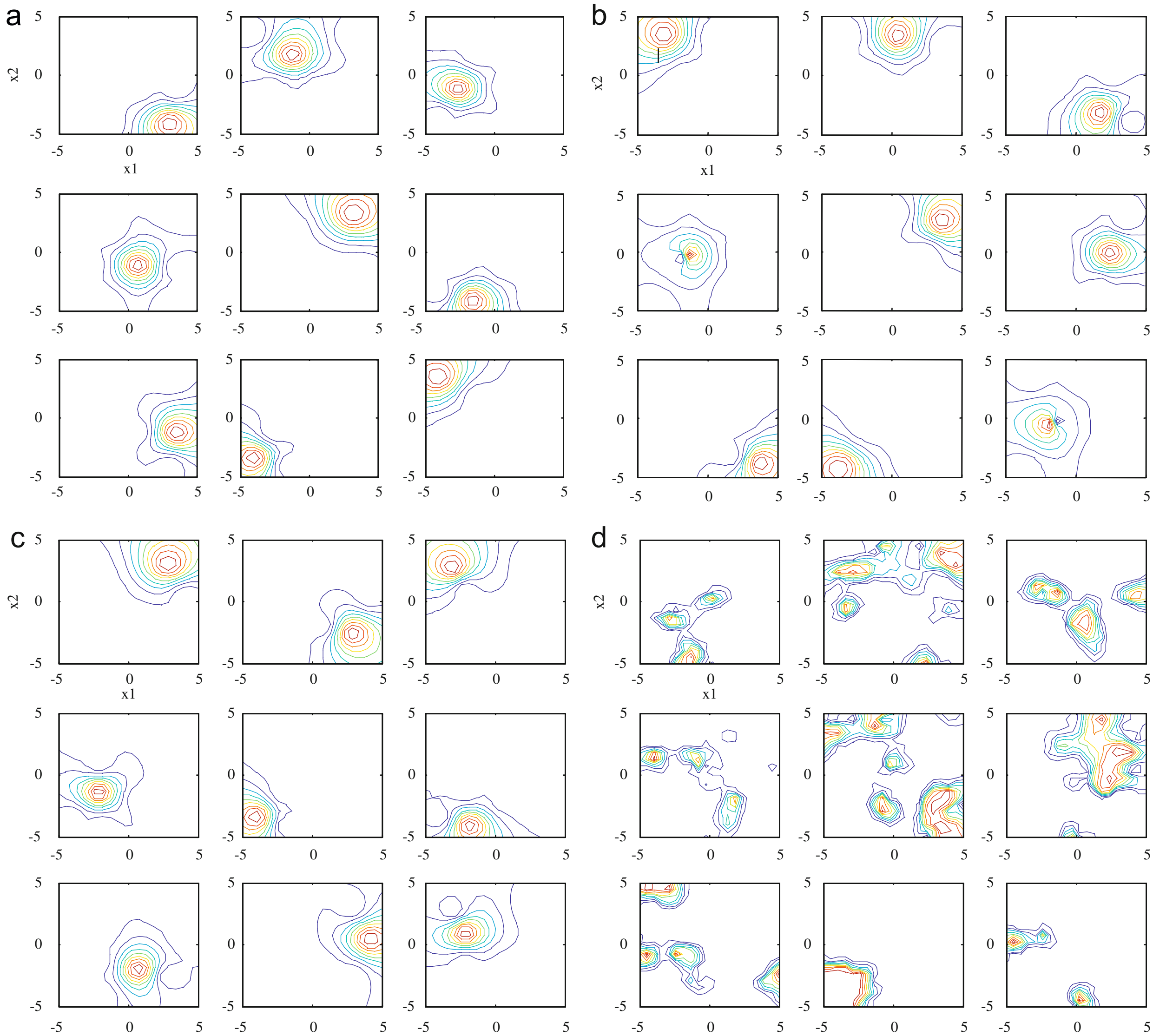

Fig. 8. Receptive fields of RBF networks obtained by four clustering methods. (a) Generic fuzzy C-means clustering; (b) supervised fuzzy clustering; (c) conditional fuzzy C-means clustering; (d) fuzzy K-NN.

and the proposed approach (i.e., fuzzy K-NN). Especially, when the conditional fuzzy C-means clustering and fuzzy K-NN have been applied, the auxiliary information granules become necessary. The information granules included in Figs. 8(c) and (d) are formed by running a generic version of the FCM shown in Fig. 7(b).

Comparing Figs. 8(a)–(d), we can see that the receptive field of RBF networks with fuzzy K-NN can be constructed as the disjoint subspaces (see the right-bottom graph in Fig. 8(d)) but on the other hand the receptive fields extracted by the conditional fuzzy C-means clustering, supervised fuzzy clustering and generic fuzzy C-means clustering are constructed as the connected spaces. This phenomenon is caused by the use of the auxiliary information granules.

Table 3 summarizes the local models z_j, $1 \le j \le 9$, and presents the values of the performance index of the associated RBFNN (shown in Figs. 8(a)–(d), respectively) with respect to the clustering methods, in the case when the constant is used as a local model of the RBFNN and the information granules are formed by the FCM.

Let us consider Fig. 8 and Table 3. Fig. 8 shows the distribution of RBFs defined by the various clustering techniques in the input space. Table 3 shows the local models which are involved in the local areas depicted by RBFs.

Table 3
Local models and the values of the performance index of the resulting RBF networks.

Clustering method	Parameter	Local model (Constant)	PI(RMSE)	EPI(RMSE)
RBFNN_FCM	$R = 9$	[1.6291, 4.2207, −1.0427, 7.7495, 3.7441, −3.7396, −1.1684, −4.1929, 3.7948]	1.6300	2.0754
RBFNN _SFC	$R = 9$	[4.2844, 7.5397, 2.3496, 0.8753, 4.0294, −3.9778, −1.11381, −1.5994, −1.0529]	1.9647	2.3816
RBFNN_CFCM	$C = 3$, $R = 3$	[0.2295, 3.8987, 3.6816, 6.9399, 7.2397, 3.3207, −5.0688, −2.4257, −6.4841]	1.3795	1.8140
RBFNN _FKNN	$C = 9$	[2.9606, 1.587, −0.7541, −3.0921, 0.5704, −5.5893, 6.0728, 4.9353, −1.7005]	0.1854	1.1063

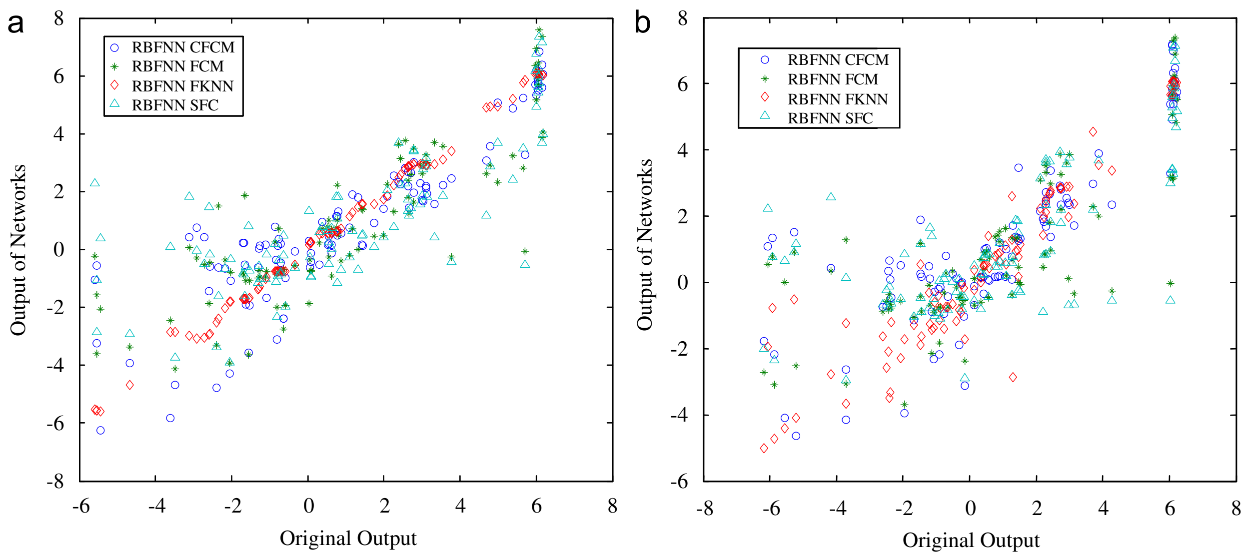

Fig. 9. Scatter plots of output data versus output of the networks (a) training data set; (b) testing data set (○: conditional fuzzy clustering, ◇: fuzzy K-NN approach, △: supervised fuzzy clustering, ∗: generic fuzzy clustering).

As mentioned above, the location of each RBF determined by clustering techniques except fuzzy K-NN is located within the connected space. The only RBFs extracted by fuzzy K-NN are disjoint receptive fields. The local models demonstrate the relationship between input variables and the output variable within the local areas defined by RBFs. In Fig. 8(d), the third local model $z_3 = -0.7541$ describes the input–output relationship within three disconnected regions which are considered as a certain RBF.

The disconnected receptive fields are considered to be quite advanced in terms of the underlying geometry supported by them. This refinement may help reduce the complexity of the local model, which in this case could be made far simpler.

Fig. 9 visualizes the approximation and generalization capabilities of the RBF networks formed with the use of various clustering methods. The results shown in Fig. 9 are produced by the RBF networks endowed with local models whose parameters are summarized in Table 3.

Table 4 summarizes the performance of the proposed model vis-à-vis other models that is RBFNN with generic fuzzy C-means clustering and RBFNN with context based clustering. In Table 4, the complexity means the complexity of the consequent local model (i.e., the number of the estimated parameters of the local model). The structure and parameters of the given model used in Table 4 stand adequate parameters which results in the best performance of the produced networks.

Table 4
Results of comparative analysis.

Model	Auxiliary information	Structure and parameters of model	PI(RMSE)	EPI(RMSE)	Complexity
RBFNN_FCM	N/A	$O = 0$, $p = 1.6$, $C = 6$	1.4181 ± 0.1592	1.7259 ± 0.1825	6
		$O = 1$, $p = 1.8$, $C = 12$	0.7200 ± 0.1257	1.1300 ± 0.1847	36
RBFNN_SFC	N/A	$O = 0$, $p = 1.2$, $C = 12$	1.6916 ± 0.2265	2.0521 ± 0.2420	12
		$O = 1$, $p = 1.8$, $C = 12$	0.8129 ± 0.1438	1.1975 ± 0.1536	36
RBFNN_CFCM	HCM	$O = 0$, $p = 1.6$, $C = 3$, $R = 6$	0.976 ± 0.140	1.396 ± 0.271	18
		$O = 1$, $p = 1.8$, $C = 3$, $R = 5$	0.591 ± 0.153	1.163 ± 0.106	45
	FCM	$O = 0$, $p = 1.4$, $C = 3$, $R = 6$	0.970 ± 0.129	1.306 ± 0.218	18
		$O = 1$, $p = 1.8$, $C = 3$, $R = 6$	0.357 ± 0.125	1.045 ± 0.385	54
	FCRM	$O = 0$, $p = 1.6$, $C = 3$, $R = 6$	0.911 ± 0.179	1.381 ± 0.219	18
		$O = 1$, $p = 1.8$, $C = 3$, $R = 6$	0.466 ± 0.085	1.150 ± 0.284	54
Proposed model (RBFNN_FKNN)	HCM	$O = 0$, $p = 2.8$, $C = 8$, $K = 6$	0.1070 ± 0.0248	0.9224 ± 0.1251	8
		$O = 1$, $p = 1.6$, $C = 7$, $K = 6$	0.090 ± 0.032	0.943 ± 0.130	21
	FCM	$O = 0$, $p = 2.0$, $C = 6$, $K = 6$	0.3126 ± 0.0350	0.9226 ± 0.1399	6
		$O = 1$, $p = 1.6$, $C = 6$, $K = 4$	0.318 ± 0.018	0.966 ± 0.188	18

Table 4 (*contd*.)

FCRM	$O = 0$, $p = 1.8$, $C = 7$, $K = 4$	0.280 ± 0.019	0.917 ± 0.165	7
	$O = 1$, $p = 1.6$, $C = 6$, $K = 4$	0.334 ± 0.062	0.973 ± 0.176	18

Table 5
Result of comparative analysis with constant local models ($O = 0$).

Data	Model	Auxiliary information	Parameters	PI(RMSE)	EPI(RMSE)	Complexity[a]
Automobile MPG data	RBFNN_FCM	N/A	$p = 1.2$, $R = 12$	3.5374 ± 0.1074	3.5360 ± 0.1835	12
		N/A	$p = 3.2$, $R = 9$	3.9683 ± 0.1435	4.3152 ± 0.4073	9
	RBFNN_SFC	N/A	$p = 1.6$, $R = 11$	3.7118 ± 0.1056	3.6886 ± 0.2767	11
		N/A	$p = 3.2$, $R = 9$	4.1723 ± 0.3732	4.1970 ± 0.2688	9
	RBFNN_CFCM	FCM	$p = 1.2$, $C = 3$, $R = 5$	3.1144 ± 0.1800	3.1893 ± 0.1588	15
		HCM	$p = 3.2$, $C = 3$, $R = 3$	3.074 ± 0.213	3.405 ± 0.273	9
	RBFNN_FKNN	HCM	$p = 3.2$, $C = 9$, $K = 5$	0.2144 ± 0.1727	2.7658 ± 0.2835	9
Boston housing data	RBFNN_FCM	N/A	$p = 1.4$, $R = 11$	6.1764 ± 0.2959	5.9963 ± 0.3362	11
		N/A	$p = 1.6$, $R = 5$	7.7894 ± 0.3689	7.5599 ± 0.4341	5
	RBFNN_SFC	N/A	$P = 1.2$, $R = 10$	6.143 ± 0.3823	6.1993 ± 0.4934	10
		N/A	$P = 1.6$, $R = 5$	7.7098 ± 0.3174	7.9405 ± 0.4326	5
	RBFNN_CFCM	HCM	$p = 2.2$, $C = 3$, $R = 6$	4.6278 ± 0.2528	4.9303 ± 0.4902	18
		FCRM	$p = 1.6$, $C = 3$, $R = 2$	5.3558 ± 0.1897	5.3204 ± 0.4813	6
	RBFNN_FKNN	FCRM	$p = 1.6$, $C = 5$, $K = 7$	1.2990 ± 0.0337	3.8948 ± 0.4985	5
CPU performance data	RBFNN_FCM	N/A	$p = 1.4$, $R = 10$	57.582 ± 17.596	78.786 ± 32.675	10
		N/A	$p = 3.6$, $R = 6$	119.966 ± 24.673	129.561 ± 38.986	6
	RBFNN_SFC	N/A	$P = 4.0$, $R = 8$	175.966 ± 14.946	130.993 ± 27.659	8
		N/A	$p = 3.6$, $R = 6$	150.884 ± 8.732	173.297 ± 11.672	6
	RBFNN_CFCM	FCRM	$p = 1.4$, $C = 2$, $R = 5$	46.249 ± 14.083	60.092 ± 22.242	5
		HCM	$p = 3.6$, $C = 3$, $R = 2$	53.160 ± 6.623	89.560 ± 25.236	6
	RBFNN_FKNN	HCM	$p = 3.6$, $C = 6$, $K = 2$	5.3838 ± 1.3178	47.116 ± 14.823	6

[a] The number of the estimated parameters of the consequent local models.

4.3. Machine learning data

In this section, we report on experiments when using some machine learning data sets [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. For evaluation purposes we selected three data sets, that is, automobile MPG data set, Boston housing data, and CPU performance data.

(1) *Automobile MPG (miles per gallon) data*: This data set deals with automobile MPG prediction. The automobile's fuel consumption is considered as an output variable that depends upon six attributes including such essential parameters as displacement, horsepower, weight, and acceleration.

(2) *Boston housing data*: This data set concerns prices of real estate in the Boston area. The MEDV (median value of the price of the house) depends on 13 continuous attributes and one binary attribute.

(3) *CPU performance data*: This data set deals with data describing a relative CPU performance. Each data point characterizes a published relative performance of the CPU being expressed in terms of its six inputs (attributes).

Table 6
Result of comparative analysis when local models are linear functions ($O = 1$).

Data	Model	Auxiliary information	Parameters	PI(RMSE)	EPI(RMSE)	Complexity[a]
Automobile MPG data	RBFNN_FCM	N/A	$p = 1.8$, $R = 3$	2.7139 ± 0.1564	2.8380 ± 0.2481	24
		N/A	$p = 2.6$, $R = 4$	2.5637 ± 0.1138	2.9922 ± 0.1897	32
	RBFNN_SFC	N/A	$P = 3.0$, $R = 3$	2.7233 ± 0.1497	2.8555 ± 0.2029	24
		N/A	$p = 2.6$, $R = 4$	2.4865 ± 0.1474	3.1877 ± 0.2826	32
	RBFNN_CFCM	FCRM	$p = 3.0$, $C = 2$, $R = 2$	2.551 ± 0.083	2.810 ± 0.195	32
			$p = 2.6$, $C = 2$, $R = 2$	2.432 ± 0.132	3.006 ± 0.203	32
	RBFNN_FKNN	FCRM	$p = 2.6$, $C = 4$, $K = 5$	1.213 ± 0.108	2.642 ± 0.205	32
Boston housing data	RBFNN_FCM	N/A	$p = 2.2$, $R = 2$	3.4647 ± 0.1915	3.8737 ± 0.3839	28
		N/A	$P = 1.7$, $R = 4$	3.1991 ± 0.1871	3.9191 ± 0.2576	56
	RBFNN_SFC	N/A	$P = 1.6$, $R = 2$	3.680 ± 0.1636	3.884 ± 0.2634	28
		N/A	$P = 1.7$, $R = 4$	3.2036 ± 0.1046	4.0713 ± 0.3209	56
	RBFNN_CFCM	FCRM	$p = 1.8$, $C = 2$, $R = 2$	3.119 ± 0.139	3.829 ± 0.305	56
		HCM	$p = 1.7$, $R = 2$, $R = 2$	3.0167 ± 0.1416	3.9433 ± 0.3341	56
	RBFNN_FKNN	HCM	$p = 1.7$, $R = 4$, $K = 7$	0.671 ± 0.075	3.665 ± 0.284	56
CPU performance data	RBFNN_FCM	N/A	$p = 1.4$, $R = 2$	40.4247 ± 3.2120	56.2589 ± 19.8709	14
		N/A	$p = 2.8$, $R = 3$	43.713 ± 7.609	65.634 ± 17.194	21
	RBFNN_SFC	N/A	$p = 1.2$, $R = 6$	29.239 ± 3.266	54.923 ± 8.6	42
		N/A	$p = 2.8$, $R = 3$	51.983 ± 7.139	78.189 ± 18.279	21
	RBFNN_CFCM	HCM	$p = 2.6$, $C = 2$, $R = 2$	30.583 ± 5.798	59.119 ± 21.158	28
		FCRM	$p = 2.8$, $C = 2$, $R = 2$	34.028 ± 4.320	70.071 ± 17.316	28
	RBFNN_FKNN	FCRM	$p = 2.8$, $C = 3$, $K = 6$	13.892 ± 1.668	44.246 ± 7.700	21

[a] The number of the estimated parameters of the consequent local models.

Table 7
Comparison of mean absolute error of the proposed model and other methods (the results reported for testing data set).

Data	Proposed model	Reference model in [5]	Linear regression [29]	Decision tree [29]	Neural networks [29]
Auto-MPG	1.94	2.02	2.61	2.11	2.02
Boston housing	2.27	N/A	3.29	2.45	2.29
CPU-performance	26.6	28.6	35.5	28.9	28.7

Table 5 summarizes the performance of the proposed model with constant values as local models versus other models that is the generic RBFNN, RBFNN with supervised fuzzy clustering and RBFNN with context based clustering. The term auxiliary information and parameters used in Tables 5 and 6 stand for the clustering method considered to extract the information granules over the output space and the design parameters for the given models which results in the best performance of the produced networks. We compare the performance of the proposed networks with the other models under the condition where the complexity and parameters of each model is as similar as possible.

In comparison with the RBFNN with the constant local models, auxiliary information based RBFNNs such as RBFNN_CFCM and RBFNN_FKNN exhibit better performances both in terms of approximation and generalization capabilities. Table 6 summarizes the performance of the proposed model with linear functions regarded as local models when contrasted with other models exploited in this study. Based on the experimental findings, we conclude that auxiliary information enhances the quality of the resulting model. When investigating Tables 5 and 6, we note a positive effect of the use of the fuzzy K-NN on the quality of the fuzzy model.

Table 8
Comparison of root mean square error of the proposed model and other methods.

Data			Proposed model	GAP-RBF [9]	MRAN [9]	RANEKF [9]	RAN [9]
Auto-MPG	Training error	Mean	0.0097	0.1144	0.1086	0.1088	0.2923
		STD	0.0037	0.0132	0.0100	0.0117	0.0808
	Testing error	Mean	0.0694	0.1404	0.1376	0.1387	0.3080
		STD	0.0091	0.0270	0.0226	0.0289	0.0915
Boston housing	Training error	Mean	0.0240	0.1507	0.1440	0.1328	0.3449
		STD	0.0003	0.0128	0.0108	0.0086	0.0620
	Testing error	Mean	0.0569	0.1418	0.1356	0.1437	0.3432
		STD	0.0102	0.0466	0.0411	0.0464	0.0770

Table 7 includes the results of comparison of the proposed model with some other methods used in the literature (cf. [5,29]). The performance of the model is expressed in terms of mean absolute error (*MAE*)

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{23}$$

For the experiments, we use a 10-fold cross validation method. The data set is divided into 10 sub-groups with nearly equal size. For each case, a sub-group is selected as the test data set and the reminders are used as the training data set which is used to train the model. The results described in Table 7 are averaged over 10 runs.

From now on, the input attributes and the output variable are normalized to the range [0, 1] for all experiments to compare the performance of the proposed model with other methods studied earlier and reported in the literature [9].

Table 8 shows the comparison analysis between the proposed model and the other model such as growing and pruning RBF (GAP-RBF), MARN, RANEKF, and RAN.

For automobile MPG data set, the entire data set is divided into 320 training data and 78 testing data randomly. In the case of Boston housing data set, the whole data set is divided into 481 training data and 25 testing data randomly.

In these experiments, the performance index is measured by the RMSE defined as given in (20).

5. Conclusions

In this study, we have proposed and investigated a new design methodology of fuzzy models based on the use of the fuzzy K-NN which supports a formation of receptive fields realized in the presence of some auxiliary supervisory information. The supervision mechanism is provided by some other clustering technique applied to data present in the output space. Such additional information granules spanned over the output space help the fuzzy K-NN develop receptive fields in the multivariable input space without loss of homogeneity for the clustered data patterns with respect to the auxiliary information over output space. As shown experimentally (the result which is also intuitively appealing), the homogeneity of data involved in any rule when evaluated with respect to the output space, is helpful in enhancing the quality of the resulting model. In the suite of experiments we investigated and quantified the impact coming from the number of rules and the values of the fuzzification coefficient. In general, we have achieved substantially lower values of the generalization errors than those already reported in the literature.

The comprehensive experimental studies which involved low-dimensional data and several commonly encountered machine learning data, lead us to acknowledge tangible benefits of the contextual information in the construction of fuzzy models.

Acknowledgment

This paper was supported by Joosan Scholarship Foundation of Wonkwang University in 2007.

References

[1] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, New York, USA, 1995.
[2] J.L. Castro, L.D. Flores-Hidalgo, C.J. Mantas, J.M. Puche, Extraction of fuzzy rules from support vector machines, Fuzzy Sets and Systems 158 (18) (2007) 2057–2077.
[3] T.M. Cover, Estimation by the nearest neighbor rule, IEEE Trans. Inf. Theory IT-14 (1) (1968) 50–55.
[4] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory IT-13 (1) (1967) 21–27.
[5] S. Destercke, S. Guillaume, B. Charnomordic, Building an interpretable fuzzy rule base from data using orthogonal least squares-application to a depollution problem, Fuzzy Sets and Systems 158 (2007) 2078–2094.
[6] J.A. Dickerson, B. Kosko, Fuzzy function approximation with ellipsoidal rules, IEEE Trans. Syst. Man Cybern. 26 (4) (1996) 542–560.
[7] S. Elanayar, Y.C. Shin, Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems, IEEE Trans. Neural Networks 5 (1994) 594–603.
[8] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning Data Mining, Inference, and Prediction, Springer, Berlin, 2001.
[9] G.B. Huang, P. Saratchandran, N. Sundararajan, An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks, IEEE Trans. SMC-B 34 (6) (2004) 2284–2292.
[10] J. Jang, C.T. Sun, Functional equivalence between radial basis function networks and fuzzy inference systems, IEEE Trans. Neural Networks 4 (1993) 156–159.
[11] C.-F. Juang, C.-T. Lin, An on-line self-constructing neural fuzzy inference network and its applications, IEEE Trans. Fuzzy Syst. 6 (1) (1998) 12–32.
[12] J.M. Keller, M.R. Gray, J.A. Givens Jr., A fuzzy K-nearest neighbor algorithm, IEEE Trans. Syst. Man Cybern. SMC-15 (4) (1985) 580–585.
[13] J.M. Keller, R. Krishnapuram, F.C.H. Rhee, Evidence aggregation networks for fuzzy logic inference, IEEE Trans. Neural Networks 3 (5) (1992) 761–769.
[14] E.T. Kim, et al., A new approach to fuzzy modeling, IEEE Trans. Fuzzy Syst. 5 (3) (1997) 328–337.
[15] C.C. Lee, Fuzzy logical in control systems: fuzzy logic controller, Part I, IEEE Trans. Syst. Man Cybern. 20 (2) (1990) 404–418.
[16] J.A. Leonard, M.A. Kramer, L.H. Ungar, Using radial basis functions to approximate a function and its bounds, IEEE Trans. Neural Networks 3 (1992) 624–627.
[17] C.J. Lin, An efficient immune-based symbiotic particle swarm optimization learning algorithm for TSK-type neuro-fuzzy networks design, Fuzzy Sets and Systems 159 (2008) 2890–2909.
[18] C.T. Lin, C.-S.G. Lee, Neural-network-based fuzzy logic control and decision system, IEEE Trans. Comput. 40 (12) (1991) 1320–1336.
[19] C.J. Lin, Y.J. Xu, A self-adaptive neural fuzzy network with group-based symbiotic evolution and its prediction applications, Fuzzy Sets and Systems 157 (8) (2006) 1036–1056.
[20] Z.Q. Liu, F. Yan, Fuzzy neural network in case-based diagnostic system, IEEE Trans. Fuzzy Syst. 5 (2) (1997) 209–222.
[21] J. Liu, M. Xu, Kernelized fuzzy attribute C-means clustering algorithm, Fuzzy Sets and Systems 159 (2008) 2428–2445.
[22] J.M. Mendel, Fuzzy logic systems for engineering: a tutorial, Proc. IEEE 83 (3) (1995) 345–377.
[23] J. Moody, C.J. Darken, Fast learning in networks of locally tuned processing units, Neural Comput. 1 (1989) 281–294.
[24] G.C. Mouzouris, J.M. Mendel, A single-value-QR decomposition based method for training fuzzy logic systems in uncertain environments, J. Intell. Fuzzy Syst. 55 (1997) 367–374.
[25] W. Pedrycz, Conditional fuzzy C-means, Pattern Recognition Lett. 17 (6) (1996) 625–632.
[26] W. Pedrycz, Conditional fuzzy clustering in the design of radial basis function neural networks, IEEE Trans. Neural Networks 9 (4) (1998) 601–612.
[27] W. Pedrycz, M. Reformat, Rule based modeling of nonlinear relationships, IEEE Trans. Fuzzy Syst. 5 (2) (1997) 256–269.
[28] W. Pedrycz, M. Reformat, Evolutionary fuzzy modeling, IEEE Trans. Fuzzy Syst. 11 (5) (2003) 652–665.
[29] J. Quinlan, Combining instance-based model and model-based learning, in: Proc. 10th ICML, San Matteo, CA, 1993, pp. 236–245.
[30] M. Sarkar, Fuzzy-rough nearest neighbor algorithms in classification, Fuzzy Sets and Systems 158 (2007) 2134–2152.
[31] A. Staiano, R. Tagliaferri, W. Pedrycz, Improving RBF networks performance in regression tasks by means of a supervised fuzzy clustering, Neurocomputing 69 (13–15) (2006) 1570–1581.
[32] C.-T. Sun, Rule base structure identification in an adaptive network based fuzzy inference system, IEEE Trans. Fuzzy Syst. 2 (1) (1994) 64–73.
[33] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man Cybern. 15 (1985) 116–132.
[34] G. Tsekourasa, H. Sarimveisb, E. Kavaklia, G. Bafasb, A hierarchical fuzzy-clustering approach to fuzzy modeling, Fuzzy Sets and Systems 150 (2) (2005) 245–266.
[35] M.G. Tsipouras, T.P. Exarchos, D.I. Fotiadis, A methodology for automated fuzzy model generation, Fuzzy Sets and Systems 159 (2008) 3201–3220.
[36] B.A. Whitehead, T.D. Choate, Evolving space-filling curves to distribute radial basis functions over an input space, IEEE Trans. Neural Networks 5 (1994) 15–23.