10

Fuzzy System with Positive and Negative Rules

Thanh Minh Nguyen and Q. M. Jonathan Wu
Department of Electrical and Computer Engineering, University of Windsor
Canada

1. Introduction

A typical rule in the rule base of a traditional fuzzy system contains only positive rules (weight is positive). In this case, mining algorithms only search for positive associations like “IF *A* Then do *B*”, while negative associations such as “IF *A* Then do not do *B*” are ignored.
The concept of fuzzy sets was introduced by Zadeh in 1965 as a mathematical tool able to model the partial memberships. Since then, fuzzy set theory (Zadeh, 1973) has found a promising field of application in the domain of image processing, as fuzziness is an intrinsic property of images and the natural outcome of many image processing techniques. The interest in using fuzzy rule-based models arises from the fact that they provide a good platform to deal with noisy, imprecise or incomplete information which is often handled exquisitely by the human-cognition system.
In a fuzzy system, we can generate fuzzy rule-bases of one of the following three types:
(a) Fuzzy rules with a class in the consequent (Abe & Thawonmas, 1997; Gonzalez & Perez, 1998). This kind of rule has the following structure:

$$\text{Rule } r: \text{ IF } x_1 \text{ is } A_{r1} \text{ and ... and } x_N \text{ is } A_{rN} \text{ Then } y \text{ is class } C_m \tag{1}$$

Where, $\mathbf{x}=(x_1,\ldots,x_N)$ is an *N*-dimensional pattern. A_{rn}, $n=(1,2,\ldots,N)$, is an antecedent fuzzy set, and y is the class C_m to which the pattern belongs.
(b) Fuzzy rules with a class and a rule weight in the consequent (Ishibuchi et al., 1992; Ishibuchi & Nakashima, 2001):

$$\text{Rule } r: \text{ IF } x_1 \text{ is } A_{r1} \text{ and ... and } x_N \text{ is } A_{rN} \text{ Then } y \text{ is class } C_m \text{ with } W_r \tag{2}$$

Where, W_r is the rule weight which is a real number in the unit interval [0,1].
(c) Fuzzy rules with rule weight for all classes in the consequent (Pal & Mandai, 1992; Mandai & Murthy, 1992; Ishibuchi & Yamamoto, 2005):

$$\begin{aligned} &\text{Rule } r: \text{ IF } x_1 \text{ is } A_{r1} \text{ and ... and } x_N \text{ is } A_{rN} \\ &\qquad\qquad \text{Then } y \text{ is class } C_1 \text{ with } W_{r1} \text{ and ... and } y \text{ is class } C_M \text{ with } W_{rM} \end{aligned} \tag{3}$$

Where, W_{rm}, $m=(1,2,\ldots,M)$, is a rule weight for class C_m.
From Eq.(1), Eq.(2) and Eq.(3), we can see that a typical rule in the rule-base of a fuzzy system contains only positive rules (weight is positive). This is one of the limitations of a

traditional association mining algorithm (Han, 2006). In this case, mining algorithms only search for positive associations like "IF *A* Then do *B*", while negative associations such as "IF *A* Then do not do *B*" are ignored. In addition to the positive rules, negative rules (weight is negative) can provide valuable information. For example, the negative rule can guide the system away from situations to be avoided, and after avoiding these areas, the positive rules once again take over and direct the process. Interestingly, very few papers have focused on negative association rules due to the difficulty in discovering these rules. Although some researchers point out the importance of negative associations (Brin & Silverstein, 1997), only few groups (Savasere et al., 1998; Wu et al., 2002; Teng et al., 2002) have proposed a system to mine these types of associations. This not only indicates the novelty in the usage of negative association rules, but also the challenges in discovering them.

In this chapter, we propose a new fuzzy rule-based system for application in image classification problems. A significant advantage of the proposed system is that each fuzzy rule can be represented by more than one class. Moreover, while traditional fuzzy systems consider positive fuzzy rules only, in this chapter, we focus on combining negative fuzzy rules with traditional positive ones, leading to fuzzy inference systems. This new approach has been tested on image classification problems with promising results.

2. Positive and Negative Association Rules

Fuzzy systems can be broadly categorized into two families. The first includes linguistic models based on a collection of fuzzy rules, whose antecedents and consequents utilize fuzzy values. The Mamdani model (Mamdani et al., 1975) falls into this group. The second category, based on Sugeno-type systems (Takagi & Sugeno, 1985), uses a rule structure that has fuzzy antecedents and functional consequent parts. A typical rule in the rule-base of a fuzzy system is of the "IF-Then" type, i.e., "IF *A* then do *B*", where *A* is the premise of the rule and *B* is the consequent of the rule. This type of rule is called *positive rule* (weight is positive) because the consequent prescribes something that should be done, or an action to be taken. Another type of reasoning that has not been exploited much, involves *negative rules* (weight is negative), which prescribe actions to be avoided. Thus, in addition to the positive rules, it is possible to augment the rule-base with rules of the form, "IF *A*, Then do not do *B*". Let us consider the following two fuzzy IF-Then rules:

$$\begin{aligned} &\text{Rule 1: IF } \textit{customer} \text{ is a } \textit{child} \text{ Then } \textit{he}\ \underline{\text{buys}}\ \textit{Coke} \text{ and } \textit{he}\ \underline{\text{does not buy}}\ \textit{bottled water} \\ &\text{Rule 2: IF } \textit{customer} \text{ is an } \textit{adult} \text{ Then } \textit{he}\ \underline{\text{buys}}\ \textit{Coke} \text{ and } \textit{he}\ \underline{\text{buys}}\ \textit{bottled water} \end{aligned} \tag{4}$$

In the example above, the negative rule (rule 1) guides the system away from situations to be avoided, after which, the positive rules (rule 2) take over and direct the process. Depending on the probability of such an association, marketing personnel can develop better planning of the shelf space in the store, or can base their discount strategies on correlations that can be found in the data itself. In some situations (Branson & Lilly, 1999; Branson & Lilly, 2001; Lilly, 2007), a combination of positive and negative rules can form a more efficient fuzzy system.

One of the limitations of fuzzy IF-Then rules in Eq.(4) is that the two classes (Coke, bottled water) appearing in the consequent parts of the above rules have the same degree of importance. Clearly, to help the marketing personnel develop better planning of different products (Coke, bottled water) for different customers (child, adult), we should assign different assign different weights to different classes appearing in the consequent parts of the rules.

Based on these considerations, we propose a new adaptive fuzzy system that applies to the image classification problem (Thanh & Jonathan, 2008). The main advantage of this fuzzy model is that every fuzzy rule in its rule-base can describe more than one class. Moreover, it combines both positive and negative rules in its structure. This approach is expressed by:

$$\begin{aligned}&\text{Rule } r: \text{ IF } x_{1k} \text{ is } A_{r1} \text{ and ... and } x_{Nk} \text{ is } A_{rN}\\ &\qquad \text{Then } y_{k1} \text{ is class } C_1 \text{ with } W_{r1} \text{ and ... and } y_{kM} \text{ is class } C_M \text{ with } W_{rM}\end{aligned} \tag{5}$$

Where, W_{rm}, r=(1,2,…,R), m=(1,2,…,M) is the weight of each class belonging to the rule r. We use the rule weight of the form below:

$$W_{rm} = w_{rm0} + w_{rm1}x_{1k} + ... + w_{rmN}x_{Nk} \tag{6}$$

Where, parameters w_{rml}, l=(0,1,…,N) are determined by the least squares estimator, which is discussed in detail, in the following section. R, M, K, and N denote the number of fuzzy rules, number of classes, number of patterns and dimension of patterns, respectively. Classes are denoted by $C_1,C_2,\ldots,C_M$, and the N-dimensional pattern is denoted by $\mathbf{x}_k=(x_{1k},x_{2k},\ldots,x_{Nk})$, k=(1,2,…,K).

Consider a multiple-input, multiple-output (MIMO) fuzzy system in Eq.(5), similar to Takagi-Sugeno fuzzy models (Takagi & Sugeno, 1985; Purwar et al., 2005). The m-th output of the MIMO with product inference, centroid defuzzifier and Bell membership functions is given by:

$$y_{km} = \frac{\sum_{r=1}^{R}\prod_{n=1}^{N} A_{rn}(x_{nk})W_{rm}}{\sum_{r=1}^{R}\prod_{n=1}^{N} A_{rn}(x_{nk})} = \frac{\sum_{r=1}^{R}\beta_r(\mathbf{x}_k)W_{rm}}{\sum_{r=1}^{R}\beta_r(\mathbf{x}_k)} = \sum_{r=1}^{R}\bar{\beta}_r(\mathbf{x}_k)W_{rm}; m = 1,...,M \tag{7}$$

Where the normalization degree of activation of the r-th rule $\bar{\beta}_r(\mathbf{x}_k)$ is expressed by:

$$\bar{\beta}_r(\mathbf{x}_k) = \frac{\beta_r(\mathbf{x}_k)}{\sum_{r=1}^{R}\beta_r(\mathbf{x}_k)}; \quad \beta_r(\mathbf{x}_k) = \prod_{n=1}^{N} A_{rn}(x_{nk}) \tag{8}$$

The fuzzy set $A_{rn}(x_{nk})$ and the corresponding rule weight W_{rm} is discussed in detail in the following section. The output of the classifier is determined by the winner-take-all strategy shown in Eq.(9), whereby "$\mathbf{x}_k$ will belong to the class with the highest activation".

$$y_k = C_{m^*}; \quad m^* = \max_{1\le m\le M}(y_{km}) \tag{9}$$

3. Structure of the proposed fuzzy system

So far, our discussion has focused on class estimation in Eq.(9) to which class the pattern $\mathbf{x}_k$ should be assigned. In this section, we suggest a new adaptive fuzzy system that can automatically adjust the values of fuzzy set $A_{rn}(x_{nk})$ and rule weight W_{rm}. After training the fuzzy system, we can determine which class the pattern $\mathbf{x}_k$ should be assigned to.

The proposed structure consists of two visible layers (input and output layer) and three hidden layers as shown in Fig. 1. This fuzzy system can be expressed as a directed graph corresponding to Eq.(7).

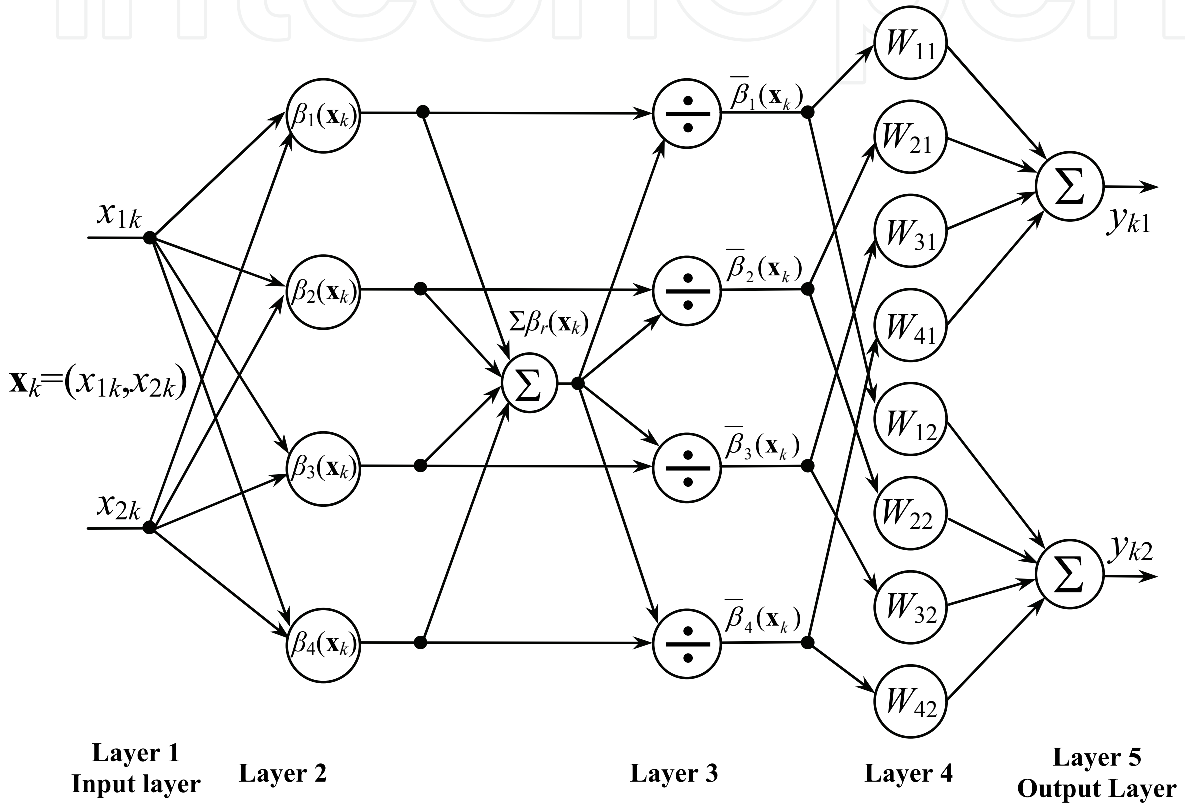


Fig. 1. Proposed fuzzy system with 2 inputs (N=2), 2 classes (M=2) and 4 rules (R=4).

Layer 1 (Input layer): each node in this layer only transmits input x_{nk}, n=(1,2,…,N), k=(1,2,…,K) directly to the next layer. No computation is performed in this layer. There are a total of N nodes in this layer, where the output of each node is $O_{1n} = x_{nk}$.

Layer 2: The number of nodes in this layer is equal to the number of fuzzy rules. Each node in this layer has N inputs from N nodes of the input layer, and feeds its output to the corresponding node of layer 3.

One of the major disadvantages of Anfis (Jang et al., 1997) model is, that an explosion in the number of inference rules limits the number of possible inputs. Thus, grid partitioning is not advised when the input dimension is more than six (Nayak et al., 2004). To overcome this problem, a fuzzy scatter partition is used in this layer. Therefore, our system can work well, even when the dimension of pattern (N) is high.

We use the bell type distribution defined over an N-dimensional pattern $\mathbf{x}_k$ for each node in this layer. The degree of activation of the r-th rule $\beta_r(\mathbf{x}_k)$ with the antecedent part $\mathbf{A}_r=(A_{r1},\ldots,A_{rN})$ is expressed as follows:

$$\beta_r(\mathbf{x}_k) = \prod_{n=1}^{N} A_{rn}(x_{nk}) = \prod_{n=1}^{N} \frac{1}{1+\left(\dfrac{x_{nk}-c_{rn}}{a_{rn}}\right)^{2b_{rn}}} \tag{10}$$

Where, parameters a_{rn}, b_{rn}, c_{rn}, $r=(1,2,\ldots,R)$, $n=(1,2,\ldots,N)$ are constants that characterize the value of $\beta_r(\mathbf{x}_k)$. The optimal values of these parameters are determined by training, which is discussed in the next section. There are R distribution nodes in this layer, where each node has 3xN parameters. The output of each node in this layer is $O_{2r} = \beta_r(\mathbf{x}_k)$.

Layer 3: This layer performs the normalization operation. The output of each node in this layer is represented by:

$$O_{3r} = \overline{\beta}_r(\mathbf{x}_k) = \frac{\beta_r(\mathbf{x}_k)}{\sum_{r=1}^{R}\beta_r(\mathbf{x}_k)} \tag{11}$$

Layer 4: Each node of this layer represents the rule weight in Eq.(6), $W_{rm}=w_{rm0}+ w_{rm1}x_{1k}+\ldots+ w_{rmN}x_{Nk}$. Where, parameters w_{rml}, $r=(1,2,\ldots,R)$, $m=(1,2,\ldots,M)$, $l=(0,1,\ldots,N)$ are determined by least squares estimator, which is discussed in the next section.

In the proposed model, for pattern $\mathbf{x}_k$, the output of the classifier is determined by the winner-take-all strategy. Therefore, when the rule weight W_{rm} has a negative value, it will narrow the choices for class C_m (the higher the negative value of W_{rm}, the smaller the value of y_{km} in Eq.(13)). In other words, negative rule weight prescribes actions to be avoided rather than performed. The output of each node in this layer is:

$$O_{4rm} = W_{rm}O_{3r} = W_{rm}\frac{\beta_r(\mathbf{x}_k)}{\sum_{r=1}^{R}\beta_r(\mathbf{x}_k)} \tag{12}$$

There are MxR nodes in this layer, where each node has (1+N) parameters.

Layer 5 (Output layer): Each node in the output layer determines the value of y_{km} in Eq.(7).

$$O_{5m} = y_{km} = \sum_{r=1}^{R} O_{4rm} = \sum_{r=1}^{R}\frac{W_{rm}\beta_r(\mathbf{x}_k)}{\sum_{r=1}^{R}\beta_r(\mathbf{x}_k)} = \sum_{r=1}^{R}\overline{\beta}_r(\mathbf{x}_k)W_{rm} \tag{13}$$

There are M nodes in the output layer.

4. Parameter Learning

The goal of the work presented here is perform the parameterized learning to minimize the sum-squared error with respect to the parameters $\mathbf{\Theta} = [a_{rn}, b_{rn}, c_{rn}, w_{rml}]$. The objective function $E(\mathbf{\Theta})$ for all the training data-sets is defined as:

$$E\left(\mathbf{\Theta}\right)=\frac{1}{2KM}\sum_{k=1}^{K}\sum_{m=1}^{M}(y_{km}-y_{dkm})^{2} \tag{14}$$

Where, y_{km} is the output of class m obtained from Eq.(7).
For a training data pair, $\{\mathbf{x}_k,\mathbf{y}_{dk}\}$, the input is $\mathbf{x}_k=(x_{1k},x_{2k},\ldots,x_{Nk})$, $k=(1,2,\ldots,K)$, and the desired output $\mathbf{y}_{dk}$ is of the form:

$$\mathbf{y}_{dk}=(y_{dk1},y_{dk2},\ldots,y_{dkM})^{T}=\begin{cases}(1,0,\ldots,0)^{T}, & if\ \ \mathbf{x}_k\in class\ C_1\\ (0,1,\ldots,0)^{T}, & if\ \ \mathbf{x}_k\in class\ C_2\\ \ldots & \\ (0,0,\ldots,1)^{T}, & if\ \ \mathbf{x}_k\in class\ C_M\end{cases} \tag{15}$$

When the initial structure has been identified with N inputs, R rules and M classes, the fuzzy system then performs the parameter identification to tune the parameters of the existing structure. To minimize the sum-squared error $E(\mathbf{\Theta})$, a two-phased hybrid parameter learning algorithm (Jang et al., 1997; Wang et al., 1999; Wang & George Lee, 2002; Lee & Lin, 2004) is applied with a given network structure. In hybrid learning, each iteration is composed of a forward and backward pass. In the forward pass, after the input pattern is presented, we calculate the node outputs in the network layers. In this step, the parameters a_{rn}, b_{rn}, and c_{rn} in layer 2 are fixed. The parameters w_{rml} in layer 4 are identified by least squares estimator. In the backward pass, the error signal propagates from the output towards the input nodes. In this step, the w_{rml} are fixed, and the error signals are propagated backward to update the a_{rn}, b_{rn} and c_{rn} by steepest descent method. This process is repeated many times until the system converges.

Next, optimization of the parameters w_{rml} in layer 4 is performed using least-squares algorithm in the forward step. To minimize the error $E(\mathbf{\Theta})$ in Eq.(14) , we have to minimize each output-error (m-th output):

$$E_m=\sum_{k=1}^{K}(y_{km}-y_{dkm})^{2} \tag{16}$$

When the training pattern $\mathbf{x}_k$ is fed into the fuzzy system, Eq.(13) can be written as:

$$\begin{aligned}y_{km}=&\bar{\beta}_1(\mathbf{x}_k)w_{1m0}+\bar{\beta}_1(\mathbf{x}_k)w_{1m1}x_{1k}+\ldots+\bar{\beta}_1(\mathbf{x}_k)w_{1mN}x_{Nk}+\\ &\bar{\beta}_2(\mathbf{x}_k)w_{2m0}+\bar{\beta}_2(\mathbf{x}_k)w_{2m1}x_{1k}+\ldots+\bar{\beta}_2(\mathbf{x}_k)w_{2mN}x_{Nk}+\\ &\ldots\\ &\bar{\beta}_R(\mathbf{x}_k)w_{Rm0}+\bar{\beta}_R(\mathbf{x}_k)w_{Rm1}x_{1k}+\ldots+\bar{\beta}_R(\mathbf{x}_k)w_{RmN}x_{Nk}\end{aligned} \tag{17}$$

For all training patterns, we have K equations of Eq.(17). Thus, Eq.(16) can be expressed:

$$E_m=\left\|\mathrm{A}\mathrm{W}_m-\mathrm{Y}_m\right\| \tag{18}$$

Where, $\mathbf{W}_m$, $\mathbf{Y}_m$, and $\mathbf{A}$ are matrices of $((N+1)*R)$x1, Kx1, and Kx$((N+1)*R)$ respectively.

$$\mathrm{W}_m = [w_{1m0}, w_{1m1}, \ldots, w_{1mN}, \ldots, w_{Rm0}, w_{Rm1}, \ldots, w_{RmN}]^{\mathrm{T}} \tag{19}$$

$$\mathrm{Y}_m = [y_{d1m} \quad y_{d2m} \quad \cdots \quad y_{dKm}]^{\mathrm{T}} \tag{20}$$

$$\mathrm{A} = \begin{bmatrix} \bar{\beta}_1(\mathbf{x}_k) & \bar{\beta}_1(\mathbf{x}_k)x_{11} & \cdots & \bar{\beta}_1(\mathbf{x}_k)x_{N1} & \cdots & \bar{\beta}_R(\mathbf{x}_k) & \bar{\beta}_R(\mathbf{x}_k)x_{11} & \cdots & \bar{\beta}_R(\mathbf{x}_k)x_{N1} \\ \bar{\beta}_1(\mathbf{x}_k) & \bar{\beta}_1(\mathbf{x}_k)x_{12} & \cdots & \bar{\beta}_1(\mathbf{x}_k)x_{N2} & \cdots & \bar{\beta}_R(\mathbf{x}_k) & \bar{\beta}_R(\mathbf{x}_k)x_{12} & \cdots & \bar{\beta}_R(\mathbf{x}_k)x_{N2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \bar{\beta}_1(\mathbf{x}_k) & \bar{\beta}_1(\mathbf{x}_k)x_{1K} & \cdots & \bar{\beta}_1(\mathbf{x}_k)x_{NK} & \cdots & \bar{\beta}_R(\mathbf{x}_k) & \bar{\beta}_R(\mathbf{x}_k)x_{1K} & \cdots & \bar{\beta}_R(\mathbf{x}_k)x_{NK} \end{bmatrix} \tag{21}$$

Next, we apply linear least-squares algorithm (Jang et al., 1997) for each output (m-th output) to tune the parameters w_{rml}.

$$\mathrm{W}_m = (\mathrm{A}^{\mathrm{T}}\mathrm{A})^{-1}\mathrm{A}^{\mathrm{T}}\mathrm{Y}_m \tag{22}$$

After the forward pass in the learning, error signals are propagated backward to update the premise parameters a_{rn}, b_{rn} and c_{rn} by gradient decent with the error function $E(\mathbf{\Theta})$ in Eq.(14). The learning rule is given by:

$$a_{rn}^{new} = a_{rn}^{old} - \eta\frac{\partial E(\mathbf{\Theta})}{\partial a_{rn}}; b_{rn}^{new} = b_{rn}^{old} - \eta\frac{\partial E(\mathbf{\Theta})}{\partial b_{rn}}; c_{rn}^{new} = c_{rn}^{old} - \eta\frac{\partial E(\mathbf{\Theta})}{\partial c_{rn}} \tag{23}$$

Where, η is the learning rate. The formulae used to update the parameters a_{rn}, b_{rn} and c_{rn} are given in the Appendix.

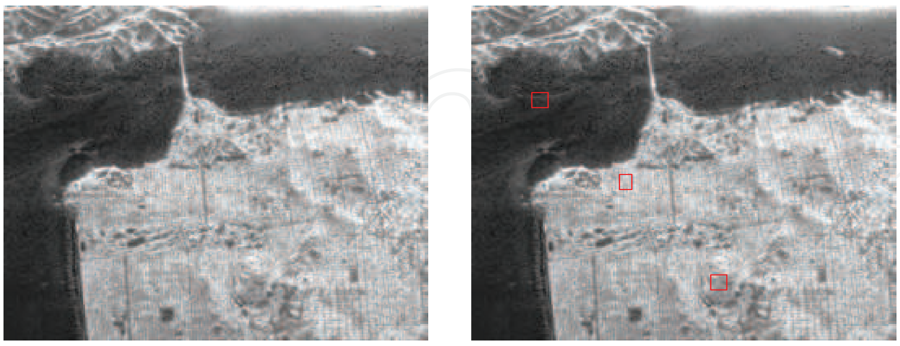
5. Simulation Results

In the first set of simulations, the proposed method is compared with Fuzzy C-Means (Hppner et al., 1999), K-Means algorithm (Dubes, 1993), Feedforward Backpropagation Network (Schalkoff & Robert, 1997; Russell et al., 2003) and Anfis methods (Jang, 1991; Jang, 1993; Russell et al., 1997). The performance of our classifier system is demonstrated for SAR Image and a natural image.

To test the effectiveness of our proposed method, in the next set of simulations, fuzzy system is used to detect the edges of the image when it is significantly degraded by high noise. The proposed system is compared with other edge-detection methods: Prewitt (Prewitt, 1970), Roberts (Roberts, 1965), LoG (Marr & Hildreth, 1980), Sobel (Sobel, 1970), and Canny (Canny, 1986).

5.1. SAR Image Classification

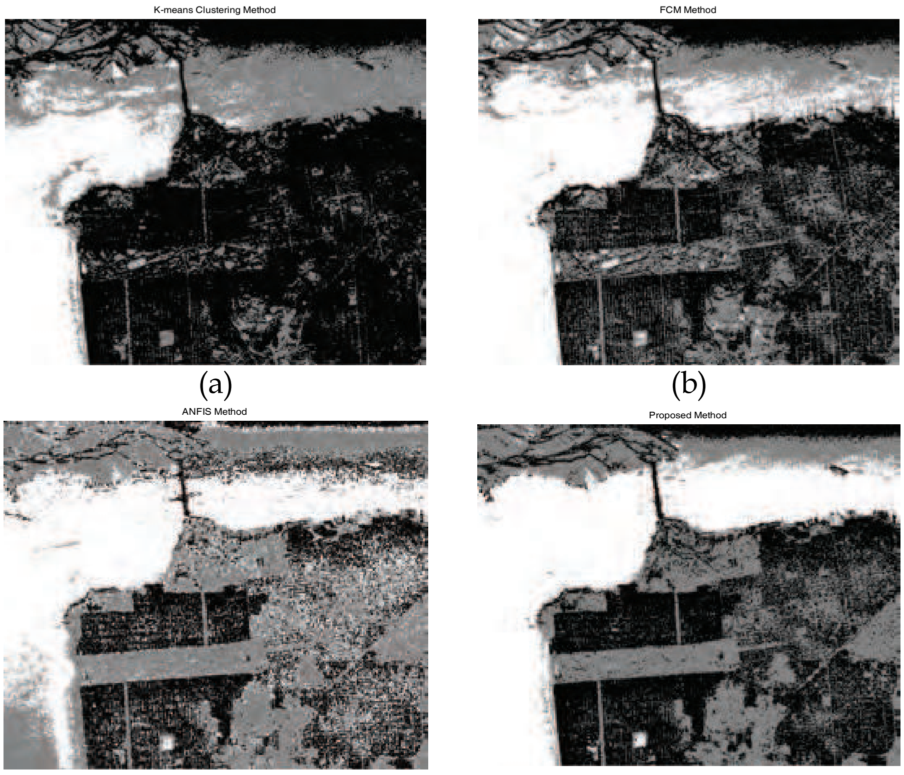
The JPL L-band polarimetric SAR image (size: 1024x900 pixels) of San Francisco Bay (Tzeng & Chen, 1998; Khan & Yang, 2005; Khan et al., 2007) as shown in Fig. 2(a) is used for this simulation. The goal is to train the fuzzy system to classify three different terrains in this image, namely *water*, *park* and *urban* areas.



(a) (b)

Fig. 2. SAR Image Classification, (a): original Image, (b): training data with 3 classes

The training patterns are shown enclosed in red boxes in Fig. 2(b). The proposed system was trained using these features to estimate the parameters. The algorithm was run with 100 training iterations.



(a) (b)

(c) (d)

Fig. 3. SAR image classification results, (a): K-Means clustering method, (b): Fuzzy C-Means methods, (c): Anfis method, (d): proposed method.

In this example, the proposed system was used to indicate three distinct classes (*M*=3), with 3 inputs corresponding to 3 polarimetric channels: *hh*, *vv*, and *vh* (Tan et al., 2007), 4 rules

(*R*=4). The desired outputs for *urban*, *park* and *water* classes were chosen to be [0 0 1], [0 1 0], and [1 0 0], respectively. After training with the patterns, the system was used to classify the whole image. Fig. 3(d) shows the classification results of the proposed method. A comparison of the proposed classifier with the K-Means classifier and Fuzzy C-Means classifier is shown in Fig. 3(a) and Fig. 3(b), respectively. These two methods were executed using MATLAB with the same 3 inputs (*hh*, *vv*, and *vh*), 3 outputs and default values for auxiliary parameters. As can be seen from Fig. 3, the classification accuracy of K-Means and Fuzzy C-Means methods was lower in water and park regions, as compared to the proposed method.

Fig. 3(c) shows the simulation result of Anfis. In this example, the same training areas in red boxes as shown in Fig. 2(b) were used to train the Anfis system. Anfis system with 3 inputs and 8 rules was run for 100 training iterations. The desired outputs for urban, park and water classes were chosen to be 1, 2, and 3, respectively. Compared with the Anfis method, clearly, our classifier accuracy is higher and the effect of noise on the performance of the detector is much less.

5.2. Natural Image Classification

In this experiment, the proposed system is compared to other classification algorithms by testing them on natural image taken from the Berkeley Dataset (Berkeley Dataset, 2001), as shown in Fig. 4.

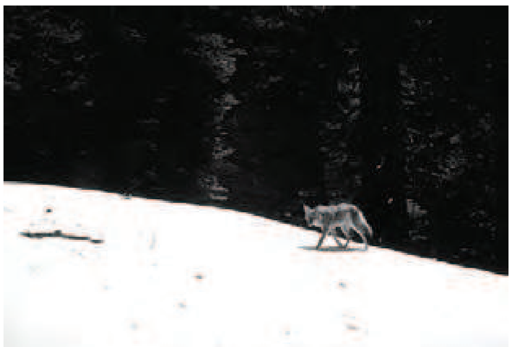


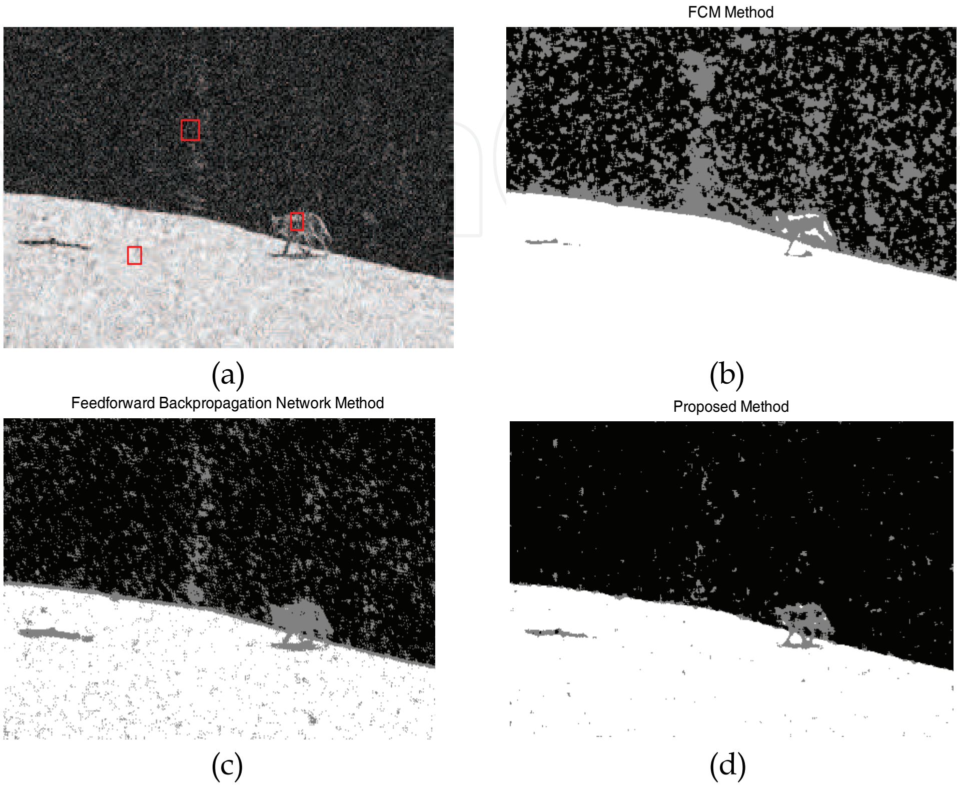
Fig. 4. Natural Image Classification.

Fig. 5(a) shows the image corrupted by Gaussian noise (0 mean, 0.1 variance) that we want to segment into 3 classes (*snow*, *wolf*, and *tree*). This input image is scanned left-to-right by taking a square window of size 5x5 pixels around a centre pixel, which is then feed into the trained fuzzy system for classification into *snow*, *wolf* or *tree*.

To train our proposed system, the training patterns are generated as shown by red boxes in Fig. 5(a). For this experiment, we have chosen a fuzzy system with 25 inputs (corresponding to the 5x5 window), 8 rules (*R*=8) and 3 distinct classes (*M*=3) with the desired outputs for *snow*, *wolf* and *tree* classes as [0 0 1], [0 1 0], and [1 0 0], respectively. Fig. 5(b) shows the clustering results of Fuzzy C-Means classifier with 25 inputs, and 3 outputs.

The image shown in Fig. 5(c) is the result obtained using Feedforward Backpropagation networks. In this example, the networks is established with the structure of 25-8-8-8-3, five layer network with 3 hidden layers, 8 neurons in each hidden layer and 3 neurons in the output layer. We use *tansig* for hidden layers and *purelin* for the output layer. Both Fuzzy C-Means and Feedforward Backpropagation networks in this example were executed using

MATLAB with default values for auxiliary parameters. As can be seen, compared to other methods, the proposed system as shown in Fig. 5(d) could not only successfully segment the image when it is significantly degraded by high noise, but also reduces the effect of noise on the final segmented image.

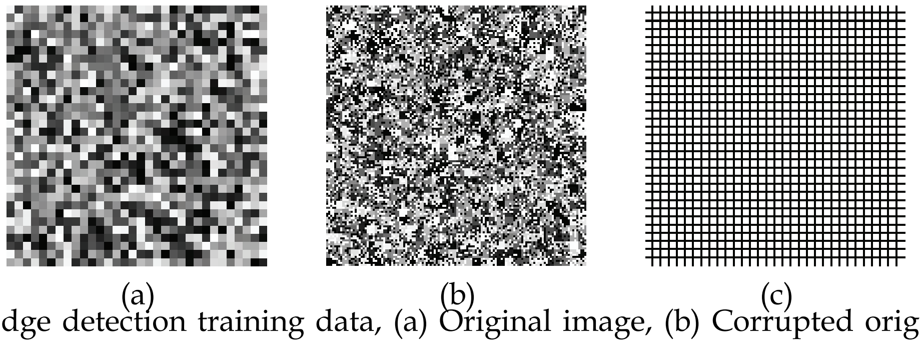


(a) (b)

(c) (d)

Fig. 5. Natural image classification results, (a): Noisy image, (b): Fuzzy C-Means methods, (c): Feedforward Backpropagation Network, (d): proposed method.

5.3. Edge Detection in Noisy Images



(a) (b) (c)

Fig. 6. Edge detection training data, (a) Original image, (b) Corrupted original image with 40% salt and pepper noise, (c) Target image.

In principle, edge detection is a two-class image classification problem where each pixel in the image is classified as either a part of the *background* or an *edge*. For this reason, a fuzzy system consisting of 2 output nodes corresponding to the 2 classes (*edge*, *background*) is chosen. In this experiment, a window of size 3x3 is scanned left-to-right across an image taken from the training set, and a determination is made as to whether the centre pixel within the square neighbourhood window belongs to an *edge* (desired output classification [0,1]) or the *background* (desired output classification [1,0]). The the fuzzy model is structured with 9 inputs (*N*=9) corresponding to the 3x3 window, 16 rules (*R*=16), and 300 training epochs, to predict the binary decision class.

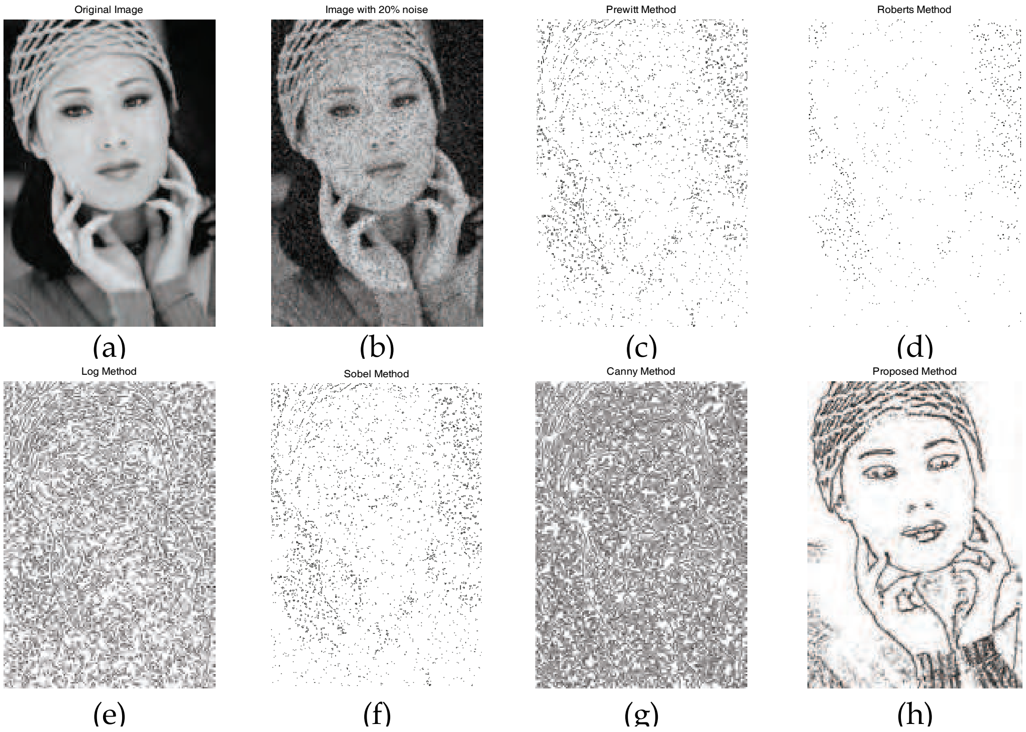


Fig. 7. The first natural image for checking (a) Original image, (b) Corrupted with 20% salt-and-pepper noise, (c) Prewitt method, (d) Roberts method, (e) LoG method, (f) Sobel method, (g) Canny method, (h) proposed method.

To train the proposed system, simple images (see Fig. 6) of size 128x128 pixels are utilized (Yksel, 2007). Fig. 6(a) shows the original image, where each square box of size 4x4 pixels has the same random luminance value. The input to the fuzzy system consists of the corrupted original image with 40% salt and pepper noise, as shown in Fig. 6(b). The target image shown in Fig. 6(c) is a black and white image, with black pixels indicating the locations of true edges in the input training image.

Once trained, the model is tested by applying it to a set of natural images taken from the Berkeley Dataset (Berkeley Dataset, 2001) as shown in Fig. 7(a). Images are corrupted with 20% of “salt” (with value 1) and “pepper” (with value 0) noise with equal probability, as shown in Fig. 7(b). The proposed detector is then compared to the existing methods - Prewitt, Roberts, LoG, Sobel and Canny detector. It is not an easy task to select good threshold values for these methods. In this case, all these methods are executed using MATLAB and with default values for auxiliary parameters. It can be easily seen that most of the edge structures of the noisy image cannot be detected by Prewitt in Fig. 7(c), Roberts in Fig. 7(d), LoG in Fig. 7(e), Sobel in Fig. 7(f) and Canny in Fig. 7(g). Besides, the effect of noise is still clearly visible as real edges are significantly distorted by the noise, and many noise pixels are incorrectly detected as edges. Comparing the results with these operators, the proposed method’s classification accuracy as shown in Fig. 7(h) is quite high, the effect of noise on the performance of the detector is much less, and the edges in the input images are successfully classified. These results indicate that the proposed system performs well when the even when image quality is significantly degraded by high noise.

Error! Reference source not found. shows the edge images which have been detected by our proposed system with different percentages of salt and pepper noise as applied to various natural images. The proposed fuzzy model consists of 16 rules (*R*=16) and 250 training epochs. The 1-st, 2-nd and 4-th column show the original images, images corrupted by 10%, and 20% salt and pepper noise, respectively. The final edge images corresponding to these noisy images as detected by the proposed system have been shown in 3-rd and 5-th columns. It can be easily seen that the proposed fuzzy system is highly robust with respect to noise in the natural images.

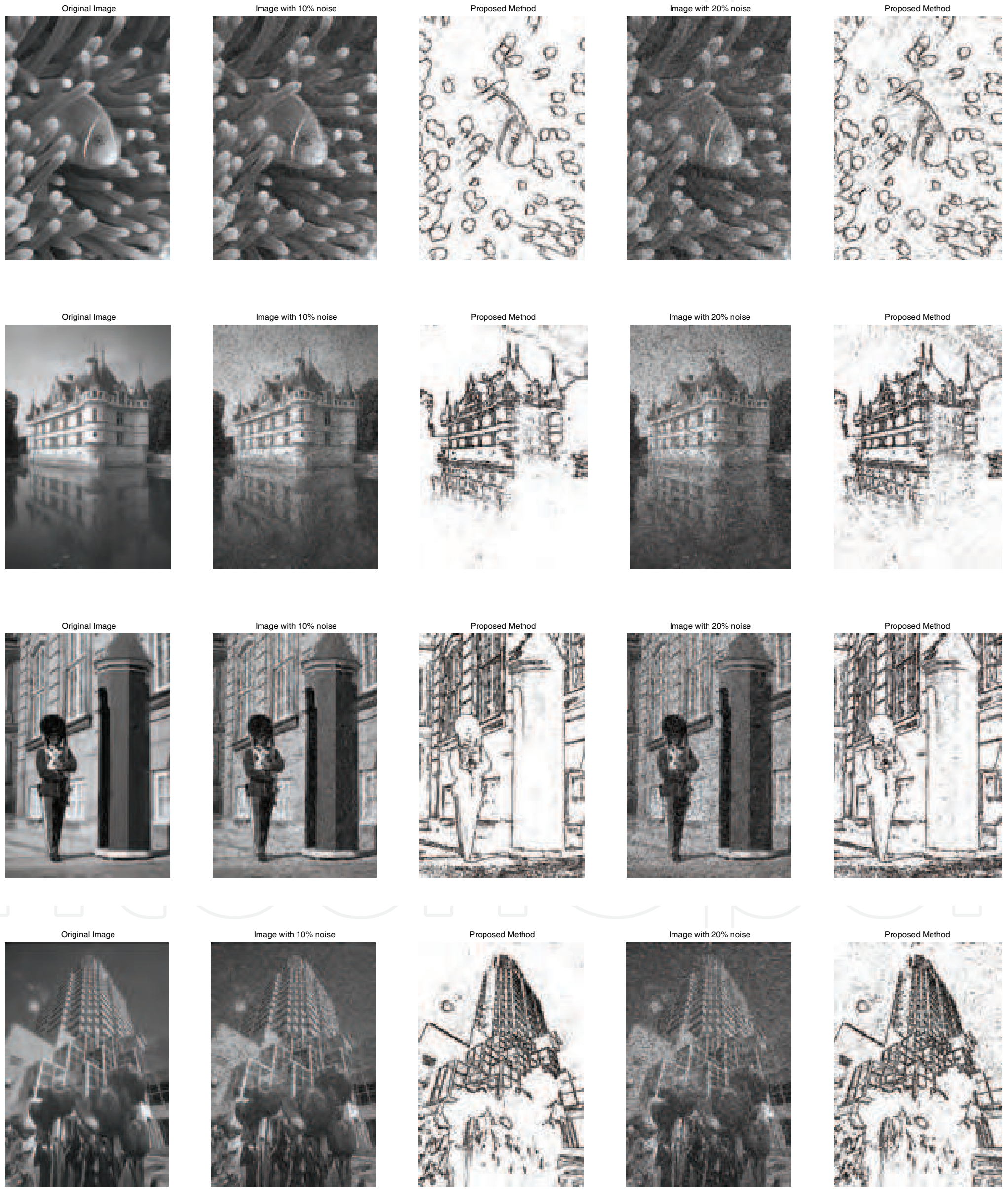


Fig. 8. The edge images which have been detected by proposed system with difference salt and pepper noise of difference natural images.

6. Conclusions

In this chapter, we have introduced a fuzzy rule-based system that combines both positive and negative association rules in its structure. A major advantage of this system is that each rule can represent more than one class. Through experimental tests and comparisons with existing algorithms on a number of natural images, it is found that the proposed system is a powerful tool for image classification.

7. Acknowledgement

This research has been supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

8. References

Abe, S. & Thawonmas, R. (1997). A fuzzy classifier with ellipsoidal regions, *IEEE Transactions on Fuzzy Systems 5 (3)*, page(s): 358-368.

Berkeley Dataset, (2001): http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench.

Branson, J.S. & Lilly, J.H. (1999). Incorporation of negative rules into fuzzy inference systems, Decision and Control, *Proceedings of the 38th IEEE Conference on*, Volume: 5, page(s): 5283-5288.

Branson, J.S. & Lilly, J.H. (2001). Incorporation, characterization, and conversion of negative rules into fuzzy inference systems, *IEEE Trans. Fuzzy Syst.*, vol. 9, page(s): 253–268.

Brin, S.; Motwani, R. & Silverstein, C. (1997). Beyond market basket: Generalizing association rules to correlations, *Proc. of SIGMOD*, page(s): 265–276.

Canny, J. (1986). A Computational Approach to Edge Detection, IEEE *Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, page(s): 679-698.

Dubes, R.C. (1993). Cluster analysis and related issues, *Handbook of Pattern Recognition and Computer Vision*, World Scientific Publishing Co., Inc., River Edge, NJ, page: 332.

Gonzalez, A. & Perez, R. (1998). Completeness and consistency conditions for learning fuzzy rules, *Fuzzy Sets and Systems 96*, page(s): 37-51.

Han, J. (2006). Learning Fuzzy Association Rules and Associative Classification Rules, Fuzzy Systems, *IEEE International Conference on*, page(s): 1454-1459.

Hppner, F.; Klawonn, F.; Kruse, R & Runkler, T. (1999). *Fuzzy Cluster Analysis*, Wiley.

Ishibuchi, H. & Nakashima, T. (2001). Effect of rule weights in fuzzy rule-based classification systems, *IEEE Trans. on Fuzzy Systems*, vol. 9, no. 4, page(s): 506-515.

Ishibuchi, H. & Yamamoto, T. (2005). Rule weight specification in fuzzy rule-based classification systems, *IEEE Trans. on Fuzzy Systems*, vol. 13, no. 4, page(s): 428-435.

Ishibuchi, H.; Nozaki, K. & Tanaka, H. (1992). Distributed representation of fuzzy rules and its application to pattern classification, *Fuzzy Sets and Systems 52*, page(s): 21-32.

Jang, J.R.; Sun, C. & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, Englewood Cliffs, NJ, page(s): 113-115.

Jang, J.S.R. (1991). Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithms, *Proc. of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, page(s): 762-767.

Jang, J.S.R. (1993). ANFIS: Adaptive-Network-based Fuzzy Inference Systems, *IEEE Trans. on Systems, Man and Cybernetics 23 (3)*, page(s): 665–685.

Khan, K. & Yang J. (2005). Novel features for polarimetric SAR image classification by neural network, *International Conference on Neural Networks and Brain*, page(s): 165-170.

Khan, K.U.; Yang J. & Zhang W. (2007). Unsupervised Classification of Polarimetric SAR Images by EM Algorithm, *IEICE Transactions 90-B(12),* page(s): 3632-3642.

Lee, C.H. & Lin, Y.C. (2004). Hybrid learning algorithm for fuzzy neuro systems, Fuzzy Systems Proceedings, *IEEE International Conference on*, Volume 2, 25-29 July 2004, page(s): 691-696.

Lilly, J.H. (2007). Evolution of a negative-rule fuzzy obstacle avoidance controller for an autonomous vehicle, Fuzzy Systems, *IEEE Transactions on, 15(4),* page(s): 718–728.

Mamdani, E.H. & Assilian S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller, *Int. J. Man-Mach. Stud.*, vol. 7, page(s): 1–13.

Mandai, D.P.; Murthy, C.A. & Pal S.K. (1992). Formulation of a multivalued recognition system, *IEEE Transactions on Systems, Man, and Cybernetics, 22 (4)*, page(s): 607-620.

Marr, D. & Hildreth E. (1980). Theory of edge detection, *Proc. of Royal Society Landon*, page(s): 187-217.

Nayak, P.C.; Sudheer, K.P.; Ragan, D.M. & Ramasastri, K.S. (2004). A neuro fuzzy computing technique for modeling hydrological time series, *Journal of Hydrology 29*, vol 291, Issues 1-2, page(s): 52-66.

Pal, S. & Mandai, D.P. (1992). Linguistic recognition system based on approximate reasoning, *Information Sciences 61*, page(s): 135-161.

Prewitt, L.G. (1970). *Object Enhancements and Extraction in Picture Processing and Psychopictorics*, Academic Press, New York, NY, page(s): 75-149.

Purwar, S.; Kar I.N. & Jha A.N. (2005). Adaptive control of robot manipulators using fuzzy logic systems under actuator constraints, *Fuzzy Sets and Systems 152*, page(s): 651–664.

Roberts, L.G. (1965). *Machine Perception of Three Dimensional Solids, in Optical and Electrooptical Information Processing*, MIT Press, Cambridge, MA, page(s): 159-197.

Russell; Stuart; Norvig & Peter (2003). *Artificial Intelligence: A Modern Approach*, 2nd Edition, Prentice Hall.

Savasere, A.; Omiecinski, E. & Navathe, S. (1998). Mining for strong negative associations in a large database of customer transactions, *Proc. of ICDE*, page(s): 494–502.

Schalkoff & Robert, J. (1997). *Artificial Neural Networks*, International Editions. McGraw-Hill.

Sobel, I. E. (1970). *Camera Models and Machine Perception*, Ph.D. Thesis, Electrical Engineering Department, Stanford University, Stanford, CA.

Takagi T. & Sugeno M. (1985). Fuzzy identification of systems and its application to modeling and control, *IEEE Trans. Syst., Man, Cybern.*, vol. 15, page(s): 116–132.

Tan, C.P.; Lim, K.S. & Ewe, H.T. (2007). Image Processing in Polarimetric SAR Images Using a Hybrid Entropy Decomposition and Maximum Likelihood (EDML), *5th International Symposium on*, Sept. 2007, page(s): 418 – 422.

Teng, W.; Hsieh, M. & Chen, M. (2002). On the mining of substitution rules for statistically dependent items, *Proc. of ICDM*, page(s): 442–449.

Thanh, M.N. & Jonathan, W.Q.M. (2008). A Combination of Positive and Negative Fuzzy Rules for Image Classification Problem, *Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications*, page(s): 741-746.

Tzeng, Y.C. & Chen, K.S. (1998). A fuzzy neural network to SAR image classification, *IEEE Trans. Geosci. Remote. Sensing*, vol. 36, page(s): 301-307.

Wang, J.S. & George Lee, C.S. (2002). Self-adaptive neuro-fuzzy inference systems for classification applications, *IEEE Trans. Fuzzy Syst*, vol. 10, page(s): 790-802.

Wang, J.S.; Lee, C.S.G. & Juang C.H. (1999). Structure and Learning in Self-Adaptive Neural Fuzzy Inference Systems, *Proc. of the Eighth Intl Fuzzy Syst. Association World Conf.*, Taipei, Taiwan, page(s): 975- 980.

Wu, X.; Zhang, C. & Zhang, S. (2002). Mining both positive and negative association rules, *Proc. of ICML*, page(s): 658–665.

Yksel, M. E. (2007). Edge detection in noisy images by neuro-fuzzy processing, *International Journal of Electronics and Communications*, vol 61, Issue 2, page(s): 82-89.

Zadeh, L.A. (1973). Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-3, no. 1, page(s): 28–44.

APPENDIX:

We apply the gradient descent technique to modify the parameters a_{rn}, b_{rn}, and c_{rn}. Parameter update formula for k-th data set of a_{rn} is represented in Eq.(24). Similarly, the update rule of c_{rn} is derived in Eq.(25). The update rule of b_{rn} is derived in Eq.(26)

$$\frac{\partial E}{\partial a_{rn}} = \frac{\partial E(\mathbf{\Theta})}{\partial y_{km}} \frac{\partial y_{km}}{\partial \bar{\beta}_r(\mathbf{x}_k)} \frac{\partial \bar{\beta}_r(\mathbf{x}_k)}{\partial \beta_r(\mathbf{x}_k)} \frac{\partial \beta_r(\mathbf{x}_k)}{\partial A_{rn}(\mathbf{x}_k)} \frac{\partial A_{rn}(\mathbf{x}_k)}{\partial a_{rn}} \tag{24}$$

$$\frac{\partial E(\mathbf{\Theta})}{\partial c_{rn}} = \frac{\partial E(\mathbf{\Theta})}{\partial y_{km}} \frac{\partial y_{km}}{\partial \bar{\beta}_r(\mathbf{x}_k)} \frac{\partial \bar{\beta}_r(\mathbf{x}_k)}{\partial \beta_r(\mathbf{x}_k)} \frac{\partial \beta_r(\mathbf{x}_k)}{\partial A_{rn}(\mathbf{x}_k)} \frac{\partial A_{rn}(\mathbf{x}_k)}{\partial c_{rn}} \tag{25}$$

$$\frac{\partial E(\mathbf{\Theta})}{\partial b_{rn}} = \frac{\partial E(\mathbf{\Theta})}{\partial y_{km}} \frac{\partial y_{km}}{\partial \bar{\beta}_r(\mathbf{x}_k)} \frac{\partial \bar{\beta}_r(\mathbf{x}_k)}{\partial \beta_r(\mathbf{x}_k)} \frac{\partial \beta_r(\mathbf{x}_k)}{\partial A_{rn}(\mathbf{x}_k)} \frac{\partial A_{rn}(\mathbf{x}_k)}{\partial b_{rn}} \tag{26}$$

Moreover, the partial derivatives in Eq.(24) to Eq.(26) are as follows:

$$\frac{\partial E(\mathbf{\Theta})}{\partial y_{km}} = \frac{1}{KM} \sum_{k=1}^{K} \sum_{m=1}^{M} (y_{km} - y_{dkm}) \tag{27}$$

$$\frac{\partial y_{km}}{\partial a_{rn}} = W_{rm} \tag{28}$$

$$\frac{\partial \bar{\beta}_r(\mathbf{x}_k)}{\partial a_{rn}} = \frac{\bar{\beta}_r(\mathbf{x}_k)}{\beta_r(\mathbf{x}_k)} \left[1 - \bar{\beta}(\mathbf{x}_k)\right] \tag{29}$$

$$\frac{\partial \beta_r(\mathbf{x}_k)}{\partial a_{rn}} = \frac{\beta_r(\mathbf{x}_k)}{1 + \left(\dfrac{x_{nk} - c_{rn}}{a_{rn}}\right)^{2b_{rn}}} \tag{30}$$

$$\frac{\partial A_{rn}}{\partial a_{rn}} = \frac{2b_{rn}}{a_{rn}} \frac{\left(\dfrac{x_{nk} - c_{rn}}{a_{rn}}\right)^{2b_{rn}}}{\left(1 + \left(\dfrac{x_{nk} - c_{rn}}{a_{rn}}\right)^{2b_{rn}}\right)^2} \tag{31}$$

$$\frac{\partial A_{rn}}{\partial c_{rn}} = \frac{2b_{rn}}{x_{nk} - c_{rn}} \frac{\left(\dfrac{x_{nk} - c_{rn}}{a_{rn}}\right)^{2b_{rn}}}{\left(1 + \left(\dfrac{x_{nk} - c_{rn}}{a_{rn}}\right)^{2b_{rn}}\right)^2} \tag{32}$$

$$\frac{\partial A_{rn}}{\partial b_{rn}} = \frac{\left(\dfrac{x_{nk} - c_{rn}}{a_{rn}}\right)^{2b_{rn}}}{\left(1 + \left(\dfrac{x_{nk} - c_{rn}}{a_{rn}}\right)^{2b_{rn}}\right)^2} \left(-\log\left(\frac{(x_{nk} - c_{rn})^{2b_{rn}}}{(a_{rn})^{2b_{rn}}}\right)\right) \tag{33}$$

Machine Learning
Edited by Yagang Zhang

ISBN 978-953-307-033-9
Hard cover, 438 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

Machine learning techniques have the potential of alleviating the complexity of knowledge acquisition. This book presents today's state and development tendencies of machine learning. It is a multi-author book. Taking into account the large amount of knowledge about machine learning and practice presented in the book, it is divided into three major parts: Introduction, Machine Learning Theory and Applications. Part I focuses on the introduction to machine learning. The author also attempts to promote a new design of thinking machines and development philosophy. Considering the growing complexity and serious difficulties of information processing in machine learning, in Part II of the book, the theoretical foundations of machine learning are considered, and they mainly include self-organizing maps (SOMs), clustering, artificial neural networks, nonlinear control, fuzzy system and knowledge-based system (KBS). Part III contains selected applications of various machine learning approaches, from flight delays, network intrusion, immune system, ship design to CT and RNA target prediction. The book will be of interest to industrial engineers and scientists as well as academics who wish to pursue machine learning. The book is intended for both graduate and postgraduate students in fields such as computer science, cybernetics, system sciences, engineering, statistics, and social sciences, and as a reference for software professionals and practitioners.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821