Generating fuzzy membership function with self-organizing feature map [☆]

Chih-Chung Yang, N.K. Bose ^{*}

*Department of Electrical Engineering, Spatial and Temporal Signal Processing Center, The Pennsylvania State University,
121 Electrical Engineering East, University Park, PA 16802, USA*

Received 24 September 2004; received in revised form 22 July 2005
Available online 18 October 2005

Communicated by A.M. Alimi

Abstract

Automatic fuzzy membership generation is important in pattern recognition. A new scheme is proposed to generate fuzzy membership functions with unsupervised learning using self-organizing feature map. Simulation results on different datasets support this new scheme. © 2005 Elsevier B.V. All rights reserved.

PACS: 07.05.Mh

Keywords: Fuzzy membership function; Pattern recognition; Self-organizing feature map; Computational intelligence

1. Introduction

The problem of fuzzy membership function generation is of fundamental importance (Medasani et al., 1998). For applications applying fuzzy set theory, one crucial part is the proper design of fuzzy membership function. For a discursive documentation, see (Cox, 1999). Many methods could be adapted to generate membership from data. These include the histogram method, transformation of probability distributions to possibility distributions, based on Zadeh's possibility theory (Zadeh, 1978), and methods based on clustering. Procedures based on clustering followed by generation of grades of membership include (Chi et al., 1995; Horikawa, 1997) using parameters (mean, variance, etc.) determined from the clusters, and fuzzy c-means (FCM) (Bezdek, 1981) algorithm capable of gener-

ating fuzzy membership function during the clustering process.

Feedforward neural network (FFNN) (Bose and Liang, 1996) can also be utilized to generate membership functions (Takagi and Hayashi, 1991; Jang and Sun, 1993) from training data with labelling. The number of neurons used in the input and output layers are the dimension of input features and the number of class labels, respectively. The desired target vector associated with the n th input feature vector is represented by a unit vector $[0 \cdots 0 \ 1 \ 0 \cdots 0]$, where 1 occurs in the c th position for encoding the labelling information of class c . After the training procedure converges with a supervised training algorithm in the learning phase, the FFNN serves as a membership generation network in the retrieving phase.

Nonrecurrent and recurrent neural network structures in conjunction with a fuzzy neural network training procedure has been used recently to classify patterns in feature space with improved capability for discriminating between patterns close to the boundaries resulting from the partitioning of the feature space (Stadter and Bose, 2000). The topology of the SOFM also has feedforward and feedback

[☆] This research was partially supported by Army Research Office Grant DAAD 19-03-1-0261.

^{*} Corresponding author. Tel.: +1 814 865 3912; fax: +1 814 865 7065.
E-mail address: nkb@stspbk.n.ee.psu.edu (N.K. Bose).

components which is exploited here to generate graded fuzzy membership during clustering process as a one-step procedure. The self-organizing feature map (SOFM) (Bose and Liang, 1996; Kohonen, 1990), which applies unsupervised learning, is often considered to be a clustering technique. However, it is also possible to extract the fuzzy membership function *directly* during the training and retrieving phases of SOFM. The proposed method achieves a similar objective in a one-step procedure that combines clustering with grade of membership generation. In Section 2, the proposed technique (PT) utilizing SOFM to elicit fuzzy membership function from data with labelling information is described. In Section 3, the technique is applied on an artificial dataset, and the well-known iris dataset. Comparison is made between the PT, a simple frequency transformation technique (referred to here as SFTT), the histogram based technique, and those using FCM and feedforward neural network. In Section 4, some insightful discussion on the proposed technique is provided.

2. New fuzzy membership function generation technique through SOFM

2.1. Self-organizing feature map (SOFM)

The SOFM is, usually, a two-layered network where the neurons in the output layer are organized into either a one- or two-dimensional lattice structure (Bose and Liang, 1996). The number of neurons in the input layer is the

dimension d of input feature vector $\mathbf{x}_n = [x_{n1} \ x_{n2} \ \dots \ x_{nd}]^T$. The synaptic weight vector at neuron j in the output layer is denoted by $\mathbf{w}_j = [w_{j1} \ w_{j2} \ \dots \ w_{jd}]^T$, $j = 1, 2, \dots, J$, where J is the total number of neurons in the output layer and w_{jk} , $k = 1, 2, \dots, d$, is the connecting weight from the j th neuron in the output layer to the k th neuron in the input layer. In the learning phase, the first step, with $\mathbf{x}_n$ as the input vector, is to find the best matching neuron from

$$q(\mathbf{x}_n) = \min_{\forall j} \|\mathbf{x}_n - \mathbf{w}_j\|, \quad (1)$$

where, for input vector $\mathbf{x}_n$, $q(\mathbf{x}_n)$ is the index label of the winning neuron $q \in \{1, \dots, J\}$, in the output layer and $\|\cdot\|$ is a distance measure (usually the Euclidean norm). The next step is to update the weight vectors associated with the label $q(\mathbf{x}_n)$. The learning rule for neuron $j \in N_q$, where N_q is the chosen neighborhood of winning neuron q for input vector $\mathbf{x}_n$, is given by

$$\mathbf{w}_j[t+1] = \mathbf{w}_j[t] + \eta_{qj}[t](\mathbf{x}_n[t] - \mathbf{w}_j[t]), \quad (2)$$

where

$$\eta_{qj}[t] = \begin{cases} \mu[t] & \text{if } j \in N_q, \\ 0 & \text{if } j \notin N_q. \end{cases} \quad (3)$$

Here, $\mu[t]$ is the learning rate, $0 < \mu[t] < 1$, at time index t .

In the retrieving phase, when $\mathbf{x}_n$ is the input vector, only the winning neuron, after convergence, will have positive response. Two different information could be retrieved from the winning neuron q , namely its index label $q(\mathbf{x}_n)$ and its associated weight vector $\mathbf{w}_q$.

Fig. 1. Proposed use of SOFM in fuzzy membership generation network: (a) learning phase and (b) retrieving phase.

2.2. Utilizing SOFM to elicit membership function from labelled data

The generation of fuzzy membership function via SOFM has, so far been a two-step procedure (Horikawa, 1997). The first step generates the proper clusters. Then, the fuzzy membership function is generated according to the clusters in the first step. However, it is possible to integrate the two-step procedure and generate the fuzzy membership function directly during the learning phase. The proposed technique is illustrated in Fig. 1. The main idea is to augment the input feature vector with the class labelling information. Similar notion can be found in (Kohonen, 1990; Mitra and Pal, 1994). In (Kohonen, 1990), the variables associated are semantic with the objective to cluster and visualize the data distribution. In (Mitra and Pal, 1994), the focus was on how SOFM could be used to handle fuzzy information. Therefore, the information being associated are all fuzzy variables.

A key step in the proposed technique is to combine the input feature vector $\mathbf{x}_n = [x_{n1} \ x_{n2} \ \dots \ x_{nd}]^T$ with the vector $\mathbf{y}_n = [y_{n1} \ y_{n2} \ \dots \ y_{nc}]^T$ coding the class labelling information. The dimensions of $\mathbf{x}_n$ and $\mathbf{y}_n$ are respectively, the number of input features d and the number of class labels c . That is, a new vector $\mathbf{z}_n$ of dimension $c + d$ is constructed according to

$$\mathbf{z}_n = [\mathbf{x}_n \ \mathbf{y}_n]^T = [\mathbf{x}_n \ 0]^T + [0 \ \mathbf{y}_n]^T. \quad (4)$$

In the learning phase, the newly constructed $\mathbf{z}_n$ will be the input feature vector to SOFM. The weight updating is according to Eqs. (1)–(3) with, in this case, the weight vector $\mathbf{w}_j = [w_{j1} \ \dots \ w_{jd} \ w_{j(d+1)} \ \dots \ w_{j(d+c)}]^T = [\mathbf{w}_{jd} \ \mathbf{w}_{jc}]$, $j = 1, \dots, J$. After the learning phase, the SOFM can be considered as a membership generation network just like its counterpart, the feedforward multilayer neural network trained with a supervised learning algorithm (Takagi and Hayashi, 1991). However, in the retrieving phase, it is not as straightforward as in the case of the feedforward multilayer neural network and some modification, described next, is required.

In the retrieving phase, the input feature vector is only $\mathbf{x}_n$. Therefore, the input feature vector will find the best matching neuron q by considering only the weight subvector $\mathbf{w}_{jd} = [w_{j1} \ \dots \ w_{jd}]^T$ related to input features, that is,

$$q(\mathbf{x}_n) = \min_{\forall j} \|\mathbf{x}_n - \mathbf{w}_{jd}\|. \quad (5)$$

After finding the winning neuron q , the output of SOFM is the weight subvector $\mathbf{w}_{qc} = [w_{q(d+1)} \ \dots \ w_{q(d+c)}]^T$, associated with the labelling information. Also, it is the fuzzy membership generated by SOFM. In order to better understand the proposed technique, an example is provided as follows.

Example. Suppose fuzzy membership functions for fuzzy variables ‘tall’ and ‘short’ in heights among a group of people are to be generated. The dataset could be collected by polling with a question “Do you think a person with height 6 feet is *tall* or *short*?” After dataset is collected, the

labelling information may be represented by 2-D unit vectors $[1 \ 0]^T$ and $[0 \ 1]^T$ for fuzzy variable ‘tall’ and ‘short’, respectively. In the training phase of SOFM, the input feature *height* was augmented with the labelling information to form a 3-D vector, which would be the input training sample for the SOFM. Suppose there are five neurons in the output layer as in Fig. 2 and the associated weights after training process are listed in Table 1. The fuzzy membership functions for the fuzzy variables *tall* and *short* are illustrated in Fig. 3.

Fig. 2. SOFM after training process.

Table 1
Weights of SOFM after training process

Neuron index	Associated weight	Feature height	Fuzzy variable	
			Tall	Short
1	$[5 \ 0 \ 1]^T$	5	0	1
2	$[5.6 \ 0.3 \ 0.7]^T$	5.6	0.2	0.8
3	$[6 \ 0.6 \ 0.4]^T$	6	0.6	0.4
4	$[6.4 \ 0.8 \ 0.2]^T$	6.4	0.8	0.2
5	$[7 \ 1 \ 0]^T$	7	1	0

Fig. 3. Fuzzy membership functions for fuzzy variables *tall* and *short*.

3. Experimental results on different datasets

In this section, the proposed technique is applied to two different datasets: an artificial dataset and the well-known iris dataset (Fisher, 1936). There are many ways to obtain the iris dataset. For example, it could be downloaded from an anonymous FTP site “ftp.ics.uci.edu” under the directory “pub/machine-learning-databases”.

3.1. Comparison of PT with SFTT

Two computer-generated datasets are designed to exemplify some basic properties of the fuzzy membership function elicited by the proposed technique. The first dataset contains two classes A and B, which have the one-dimensional (1-D) Gaussian distribution with same variance $\sigma^2 = 9$ and different mean values $\mu_A = -10$, and $\mu_B = 10$, respectively. One hundred samples are randomly generated for each class and the range $[-20, 20]$ is considered. The histogram is illustrated in Fig. 4(a). The dotted and solid lines show plots for the probability functions of Gaussian distributions for classes A and B, respectively. Two techniques applying SOFM are compared here as a starting point.

The first technique, which could be called the simple frequency transformation technique (SFTT), combines the notions of histogram and transformation of probability distributions to possibility distributions (Medasani et al., 1998). The 1-D SOFM with 10 neurons in the output layer

is trained with 200 1-D feature vectors (without labelling information). After the SOFM converges, the positive responses for all output neurons are recorded according to the class labels for all training vectors. The relative frequencies for the two classes are calculated for all 10 neurons. For example, suppose there were 20 input samples to output neuron 4. Among these 20 input samples, 16 were in class A and the rest 4 were in class B. The relative frequencies for classes A and B were $16/20 = 0.8$ and $4/20 = 0.2$, respectively. The relative frequencies are plotted according to the weights w_j associated with the 10 neurons of output layer, as in Fig. 4(b). The resulting fuzzy membership functions for classes A and B are plotted as dotted and solid lines, respectively, by connecting $\circ$ and $\times$.

For the proposed technique (PT), the class label vectors $[1\ 0]^T$ and $[0\ 1]^T$ for classes A and B, respectively, are first augmented with the associated 1-D input feature vectors. Then, the SOFM with the same topology (10 neurons in the output layer) is trained with the newly constructed 3-D input vectors (1-D input feature plus 2-D class information vector) until the SOFM converges. The weights associated with class information A and B provide the grade of membership which is plotted against the weights associated with 1-D input feature vector, as in Fig. 4(c). As before, the resulting fuzzy membership functions for classes A and B are plotted as dotted and solid lines, respectively, by connecting $\circ$ and $\times$.

In real applications, the number of data samples for different classes may not be the same. Therefore, it is

Fig. 4. Simulation results on Gaussian distribution comparing SFTT and PT.

reasonable to unbalance the training samples to observe the effects on the shape of fuzzy membership function. First, the number of training samples for class A was reduced to 50 and the histogram was shown in Fig. 4(d). As Fig. 4(a), the dotted and solid line indicates the Gaussian probability functions that generated the data samples for classes A and B, respectively. The membership functions generated by the two techniques using SOFM are shown in Fig. 4(e) for SFTT and (f) for PT. For the SFTT, the cross-over point is slightly shifted to the right in comparison with Fig. 4(b). For the PT, there are no apparent changes.

In an extreme case, the number of training samples for class A is further reduced to 10, as the histogram illustrated in Fig. 4(g). Again, the membership functions generated by the two techniques using SOFM are shown in Fig. 4(h) and (i). For the SFTT, one ‘dead’ neuron (has no response to any training data sample) emerged in the transition region between the two classes A and B. Therefore, no relative frequencies could be calculated. Therefore, no membership value was assigned to that neuron and the associated region is connected by a dash-dot line to indicated possible interpolated fuzzy values in that region. For the PT, the cross-over point slightly moves to the left and the situation of ‘dead’ neuron was avoided because there was no transformation of probability distributions to possibility distributions.

The second dataset again contains two classes A and B. However, the probability model for the random generation

of data samples is uniform distribution. The data ranges for classes A and B are $[-20, 0]$ and $[0, 20]$ with mean values at $\mu_A = -10$, and $\mu_B = 10$, respectively. Similar to the previous three experimental setups for the Gaussian dataset, the number of training samples coming from class A were 100, 50, and 10. For class B, all the 100 samples are used for training SOFM. The histograms of the training sets, which were consistent with the uniform distribution model (plotted in dotted and solid lines for classes A and B, respectively), were as shown in Fig. 5(a), (d), and (g).

Again, the fuzzy membership functions generated with the SFTT are shown in Fig. 5(b), (e), and (h), accordingly. The results generated by the PT are shown in Fig. 5(c), (f), and (i). By applying the PT, the shape of fuzzy membership is robust to unbalanced training datasets as well as different model assumption of data distribution (by comparing Figs. 4 and 5).

3.2. Comparison of PT with histogram, FCM and FFNN

In order to better exemplify the superiority of the proposed technique, three different fuzzy membership function eliciting methods were visually compared based on the same artificial datasets. The methods being compared were the histogram, the fuzzy c-means (FCM), and the feedforward neural network (FFNN). For the histogram method, the number of training samples in each bin was counted first according to the labelling information. Then, the histogram was normalized by dividing the maximum number

Fig. 5. Simulation results on uniform distribution comparing SFTT and PT.

among all bins. Therefore, the maximum fuzzy membership value would equal to 1. The resulting fuzzy membership functions were plotted in the first column (subplots (a), (d), and (g)) in Figs. 6 and 7 for Gaussian and uniform distributions, respectively. It is obvious that the shapes of the output fuzzy membership function were highly affected by the distribution and the number of training samples. This method might not be suitable for application with small data samples.

FCM (Bezdek, 1981) is a clustering method which embeds the generation of fuzzy membership function while clustering. The algorithm is based on minimization of the objective function:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \| \mathbf{x}_i - \mathbf{c}_j \|^2, \quad m > 1,$$

where m is any real number greater than 1, u_{ij} is the degree of fuzzy membership of data sample $\mathbf{x}_i$ in cluster j , $\mathbf{c}_j$ is the center of the j th cluster, and $\| \cdot \|$ is any norm expressing the similarity between any data sample and the center. An iterative optimization of the objective function is carried out through the update of membership u_{ij} ,

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\| \mathbf{x}_i - \mathbf{c}_j \|}{\| \mathbf{x}_i - \mathbf{c}_k \|} \right)^{\frac{2}{m-1}}}, \quad (6)$$

and the cluster centers $\mathbf{c}_j$,

$$\mathbf{c}_j = \frac{\sum_{i=1}^N u_{ij}^m \mathbf{x}_i}{\sum_{i=1}^N u_{ij}^m}. \quad (7)$$

This iteration will stop when certain termination criterion is met. That is, $\max_{ij} \{ |u_{ij}^{(k+1)} - u_{ij}^{(k)}| \} < \varepsilon$, where ε satisfies $0 < \varepsilon < 1$ and the superscript k denotes iteration number. In the simulation of FCM on artificial dataset, the parameters were set to be $m = 2$ and $\varepsilon = 10^{-5}$. The labelling information was used for initialization of the FCM algorithm. The resulting fuzzy membership functions were plotted in the second column (subplots (b), (e), and (h)) in Figs. 6 and 7 for Gaussian and uniform distribution, respectively. It is observed that the imbalanced data samples caused distortion in shape both for the Gaussian and uniform distribution cases (Figs. 6(h) and 7(h)).

The last method being evaluated was the FFNN method. The model being used had eight neurons in the hidden layer and two neurons in the output layer. The target vectors for classes A and B were $[1 \ 0]^T$ and $[0 \ 1]^T$, respectively. The outputs of the fuzzy membership generation network are respectively shown in the third column (subplots (c), (f), and (i)) in Figs. 6 and 7 for Gaussian and uniform distributions. It could be noticed that this method is better against the imbalance in training samples. However, the distribution of the training samples would influence the shapes of fuzzy membership functions by comparing Figs. 6 and 7.

Fig. 6. Simulation results on Gaussian distribution comparing histogram, FCM, and FFNN.

Fig. 7. Simulation results on uniform distribution comparing histogram, FCM, and FFNN.

3.3. Further comparison of PT with FCM

Extended from previous simple cases, more complicated datasets with different degrees of overlapping on class boundaries are examined in this section. In Figs. 8 and 9, three different overlap conditions based on, respectively, Gaussian and uniform distributions are shown. Again, there are two classes A and B, each of which has 100 samples. Class A histogram is inverted for demonstration purpose. The proposed method and FCM method are compared. The variance of each class is 9 in Fig. 8 and the mean values are $\{-6, 6\}$, $\{-3, 3\}$, and $\{-1.5, 1.5\}$, as shown in subplots (a), (d), and (g), respectively. The results applying our proposed method for the three different cases are shown in subplots (b), (e), and (h), respectively. The results applying FCM for three different cases are shown in subplots (c), (f), and (i), respectively.

The advantages of our proposed method are obvious. The more typical patterns (at extreme ends) have the highest fuzzy grade of membership assignment. At the transition region, depending on the relative frequency of both classes around neuron position, a proper membership value is assigned.

In Fig. 9, the data span for classes A and B are $[-17.5, 2.5]$, $[-2.5, 17.5]$, $[-15, 5]$, $[-5, 15]$, and $[-12.5, 7.5]$, $[-7.5, 12.5]$, as shown in subplots (a), (d), and (g), respectively. The respective results applying our proposed method and the FCM method are shown in subplots (b), (e), and (h), and subplots (c), (f), and (i), respec-

tively. An interesting phenomenon in this example is the flat membership value assignment in the transition region, as shown in the center parts of subplots (e) and (h). It is proper to assign equal membership value in the overlapped region, as is done by applying our proposed method, because the equal uniform distribution of the datasets have the same possibility in the overlapped center region. The model-based FCM method failed to assign proper fuzzy membership value due to its Gaussian model assumption.

3.4. Iris dataset

In the iris dataset, there were three different classes: iris Setosa, iris Versicolour, and iris Virginica. There were four measured features: sepal length, sepal width, petal length, and petal width (unit: cm). The iris dataset demonstrates that the proposed technique can be applied to multidimensional input features with multiple labelling information. The output layer of the SOFM used for this experiment has a 3×3 square grid topology for simplicity. For demonstration purpose, the four input features were grouped into two 2-D vectors $\mathbf{x}_{\text{sepal}}$ and $\mathbf{x}_{\text{petal}}$, which contain the length and width of the sepal and the petal, respectively. The target vectors $\mathbf{y}_n$ were designed to be unit vectors $[1 \ 0 \ 0]^T$, $[0 \ 1 \ 0]^T$ and $[0 \ 0 \ 1]^T$ for iris Setosa, iris Versicolour, and iris Virginica, respectively. Therefore, the input vectors $\mathbf{z}_n$ used for training SOFM were 5-D vectors. The results after training were shown in Figs. 10 and 11 for 2-D input feature vectors $\mathbf{x}_{\text{sepal}}$ and $\mathbf{x}_{\text{petal}}$, respectively. The integers

Fig. 8. Simulation results on Gaussian distribution comparing PT and FCM.

Fig. 9. Simulation results on uniform distribution comparing PT and FCM.

inside the figure are the indices for neuron $j \in \{1, \dots, 9\}$ in the SOFM output layer. The symbols $\times$, $\circ$, and $\square$ repre-

sent iris Setosa, iris Versicolour, and iris Virginica, respectively. The associated fuzzy membership for different kinds

Fig. 10. Fuzzy membership generated by the proposed technique (PT) with sepal length and width.

Fig. 11. Fuzzy membership generated by the proposed technique (PT) with petal length and width.

of iris are shown individually in the subplots (a), (b) and (c) of Figs. 10 and 11, correspondingly. The weight subvectors

of SOFM, associated with the input feature vectors, were marked as dots with index labels 1–9 in the subplot (d)

of Figs. 10 and 11. Comparing the subplots (d) with the other three fuzzy membership functions in subplots (a), (b), and (c), it could be observed that the estimation of fuzzy membership function was consistent with the distribution of patterns in feature space.

4. Discussion and conclusion

The proposed technique and experimental results offer a *neural network view (NN view)* on the meaning of fuzzy membership. Since fuzzy set theory and neural networks originate from the same model—human brain, a trend toward combining of the techniques from these two fields is natural (Stadter and Bose, 2000; Jang et al., 1997).

Unlike the supervised NN technique, a minor drawback is the discrete output due to the limited number of neurons J in the output layer of SOFM. However, this could be overcome by incorporating proper interpolation methods. The proposed technique (or, broadly speaking, the neural network driven techniques) is also suitable for generating multidimensional fuzzy variables. The advantage lies on the reduction of number of rules in the rule-based system (Takagi and Hayashi, 1991). Furthermore, the visualization capability (see the plots of the grades of membership versus neuron location in Figs. 10 and 11) of SOFM on high dimensional data is a valuable characteristic in any fast prototype design procedure.

Another important feature is the robustness of this proposed technique. In other automatic fuzzy membership function elicitation methods, the frequency of occurrence, the imbalance of dataset, or the different distribution model assumption will have impacts on the shape of membership function. From the simulation experiments reported here, it can be observed that the proposed technique is robust. It also agrees with the finding of (Hersh et al., 1979) that frequency of occurrence of the elements does not affect the location and form of the membership function. Our results also suggest that the membership function is not only a function of the object from the universe of discourse but the discourse as well (Hersh et al., 1979). Summarizing, a fuzzy membership generation technique has been proposed which fully uncovers the capability of SOFM in eliciting fuzzy membership directly from data with labelling information. In the future, this new idea

of fuzzy clustering is expected to be used on diverse datasets by different researchers.

Acknowledgement

The authors thank the three reviewers and the editor for their constructive suggestions, which have been incorporated in the several revisions of the original submission.

References

- Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.
- Bose, N.K., Liang, P., 1996. Neural Networks Fundamentals with Graphs, Algorithms, and Application, first ed. Prentice-Hall, Inc., New York, NY.
- Chi, Z., Wu, J., Yanm, H., 1995. Handwritten numeral recognition using self-organizing maps and fuzzy rules. Pattern Recognition 28 (1), 59–66.
- Cox, E., 1999. The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems, second ed. AP Professional, San Diego.
- Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugenics 7, 179–188.
- Hersh, H., Carmazza, A., Brownell, H.H., 1979. Effects of context on fuzzy membership functions. In: Gupta, M.M., Ragade, R.M., Yager, R.R. (Eds.), Advances in Fuzzy Set Theory. North-Holland, Amsterdam, pp. 389–408.
- Horikawa, S., 1997. Fuzzy classification system using self-organizing feature map. Oki Tech. Rev. 63(159), [Online]. Available from: <<http://www.oki.com/en/otr/html/nf/otr-159-05.html>>.
- Jang, J.-S.R., Sun, C.-T., 1993. Functional equivalence between radial basis functions and fuzzy inference systems. IEEE Trans. Neural Networks 4 (1), 156–159.
- Jang, J.R., Sun, C.T., Mizutani, E., 1997. Neuro-Fuzzy and Soft Computing. Prentice-Hall, Inc., Upper Saddle River, NJ.
- Kohonen, T., 1990. The self-organizing map. Proc. IEEE 78 (9), 1464–1480.
- Medasani, S., Kim, J., Krishnapuram, R., 1998. An overview of membership function generation techniques for pattern recognition. Internat. J. Approx. Reason. 19, 391–417.
- Mitra, S., Pal, S.K., 1994. Self-organizing neural network as a fuzzy classifier. IEEE Trans. Systems Man Cybernet. 24 (3), 385–399.
- Stadter, P.A., Bose, N.K., 2000. Neuro-fuzzy computing: Structure, performance measure and applications. In: Pal, S.K., Ghosh, A., Kundu, M.K. (Eds.), Soft Computing for Image Processing. Physica-Verlag, Heidelberg, pp. 337–374.
- Takagi, H., Hayashi, I., 1991. NN-driven fuzzy reasoning. Internat. J. Approx. Reason. 5, 191–212.
- Zadeh, L.A., 1978. Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets System 1, 3–28.