An overview of membership function generation techniques for pattern recognition

Swarup Medasani a, Jaeseok Kim b, Raghu Krishnapuram a,*

a Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO 80401, USA
b Samsung Biomedical Research Institute, 50 Ilwon-dong, Kangnam-Ku, Seoul, South Korea-135710

Received 1 April 1997; accepted 1 March 1998

Abstract

The estimation of membership functions from data is an important step in many applications of fuzzy theory. In this paper, we provide a general overview of several methods for generating membership functions for fuzzy pattern recognition applications. We discuss methods based on heuristics, probability to possibility transformations, histograms, nearest neighbor techniques, feed-forward neural networks, clustering, and mixture decomposition. We also illustrate these membership generation methods using synthetic and real data sets, and discuss the suitability and applicability of these membership function generation techniques to particular situations. © 1998 Elsevier Science Inc. All rights reserved.

Keywords: Membership function generation; Typicality; Probability to possibility transformations; K-nearest neighbor; Clustering; Mixture decomposition; Multi-prototype classification
1. Introduction

Eliciting membership functions from (training) data is one of the fundamental issues associated with the application of fuzzy set theory. There are no guidelines or rules that can be used to choose the appropriate membership generation technique. Another problem that makes membership function generation a non-trivial task, is the lack of consensus on the definition and interpretation of membership functions. For example, Dubois and Prade [1] point out three different interpretations of the statement: "The membership value of George Bush in the class of tall men is 0.8." The three interpretations are: (a) 80% of the population declared that George Bush is tall (likelihood view), (b) 80% of the population described "tall" as an interval containing George Bush's height (random set view), and (c) George Bush's height is at a normalized distance equal to 0.2 from the closest ideal prototype (typicality view). Thus, one could think of a variety of methods to generate membership values depending on the interpretation. Much literature is oriented toward the determination of membership functions that reflect subjective perceptions about vague or imprecise concepts such as tall persons, old men, and young men. Unfortunately, these methods cannot be directly applied to many practical problems. For example, in fuzzy logic applications, membership functions are needed to model the prevailing uncertainty in the input information. It should be noted that no measures are available to evaluate the goodness or correctness of the membership function generated using a particular method. This is a serious problem when membership functions are used to model concepts that have no physical meanings. Therefore, models used for membership functions must be sufficiently flexible so that they can be easily adjusted or tuned to optimize the performance of the algorithm that uses them.

The problem of membership function generation is of fundamental importance because the success of an algorithm depends on the membership functions used. It might be difficult, if not impossible, to come up with a single membership generation method which will work for most applications. Rather, several methods may have to be used in tandem, and the choice of the method may depend on the kind of the problem and the type of data available. Other approaches to model uncertainty (such as probability) also use a variety of methods to estimate the underlying distributions depending on the situation. In this paper, we focus on methods of membership function generation for pattern recognition applications. We suggest several methods to estimate membership functions and make some qualitative comparisons.

In Section 2, we review membership function generation techniques that reflect subjective perception. Membership functions based on heuristic functions are presented in Section 3. In Section 4, we show how multi-dimensional histograms of input features can be used for generating membership functions. In Section 5, some methods to transform probability distributions to possi-
2. Membership functions based on perception

In many decision-making applications, membership functions of fuzzy sets are based on subjective perceptions of vague or imprecise categories rather than on data or other objective entities involved in the given problem. The problem of assigning numbers to subjective perceptions of vague categories is a matter of mathematical psychology and requires the utilization of various techniques of the theory of measurement and scaling.

Extensive experiments on generating membership functions for concepts such as *tall man* and *aesthetically pleasing house* were conducted by Norwich and Turksen [2] with the assumption that membership values should be defined on an interval scale. An interval scale is justified by the inapplicability of extensive measurements to fuzziness and the lack of a natural origin for memberships. In their experiments, two techniques, called direct rating and reverse rating, were used. In the direct rating procedure, the subject is presented with a random series of persons (houses) and then asked to indicate the membership degree to rate each one as *tall* (or *pleasing*). In the reverse rating procedure, the subject is presented with an ordered series of persons (houses) and asked to select the one person (house) best seeming to correspond to the indicated degree of membership in the category of *tall* persons (or *pleasing* houses).

Another method proposed involves the polling technique [3,4]. This technique assumes that semantic uncertainty is merely a statistical uncertainty in the information-theoretic sense. The values of membership functions are found by randomly and repeatedly presenting a subject with elements and acquiring either a 'yes' or a 'no' response to the question: *Does x belong to A?* This polling method implies that probability of a positive answer is proportional to membership value. This interpretation of membership corresponds to the *likelihood view*.

Saaty [5] used the relative preference method with the assumption that membership value is on a rational scale. Membership values are computed from the matrix \( A = [a_{ij}] \), called the pairwise comparison alternative matrix, whose element \( a_{ij} \) represents the relative membership value of an element \( x_i \) in a fuzzy set \( F \) with respect to the membership value of an element \( x_j \) in \( F \). The element \( a_{ji} \) in the \( j \)th row and \( i \)th column of this matrix is \( 1/a_{ij} \). Saaty used a
scale divided into seventeen levels \( \{1/9, 1/8, \ldots, 1/2, 1, 2, \ldots, 8, 9\} \). Each of these levels has a semantic interpretation. The larger the value of \( a_{ij} \), the greater the membership of \( x_i \) compared with that of \( x_j \). The membership values are determined by finding the eigenvector of \( A \) such that \( Au = \mu u \), where \( A \) is assumed to be as consistent as possible (i.e., the matrix \( A \) really represents relative membership values), \( n \) is the number of elements in the reference set and \( u \) is the membership vector \( [u_1, u_2, u_3, \ldots, u_n]^T \).

Parametrized membership functions have been suggested by several researchers. Kochen and Badre [6] assumed that a membership function is continuous, differentiable and S-shaped. Furthermore, the marginal increase of a person's strength of belief that "\( x \) is \( A \)" is assumed to be proportional to the strength of his/her belief that "\( x \) is \( A \)" and to the strength of his/her belief that "\( x \) is not \( A \)". That is,

\[
\frac{du(x)}{dx} = ku(x)(1 - u(x))
\]

and its solution is given by

\[
u(x) = \frac{1}{1 + \exp(a - bx)}. \tag{1}\]

This can be easily linearized since \( \ln((1 - u(x))/u(x)) = a - bx \), and the parameters \( a \) and \( b \) can be determined by statistical methods such as regression. A similar membership function was derived by Zimmermann and Zysno [7]. Their parametrization of a membership function is

\[
u(x) = \frac{1}{1 + d(x)}. \tag{2}\]

Membership is defined as a function of the distance \( d(x) \) between a given object \( x \) and a standard (ideal) member. This corresponds to the typicality view discussed in Section 1. Hence, \( d(x) = 0 \Rightarrow u = 1; d(x) = \infty \Rightarrow u = 0 \). The distance function \( d(x) \) can be \( 1/x \), but, due to the evidence that the relationship between a physical unit and perception is generally exponential, a more appropriate distance function is \( 1/\exp(-a(x - b)) \) where the parameter \( a \) models the slope of the membership function and \( b \) represents the point at which the tendency of the subject's attitude changes from being rather positive to being rather negative (inflection point).

Another parametrized membership function suggested by Dombi [8] is given by

\[
u(x) = \frac{(1 - v)^{1-\lambda}(x - a)^\lambda}{(1 - v)^{1-\lambda}(x - a)^\lambda + v^{\lambda-1}(b - x)^\lambda}, \quad \lambda > 1,
\]

where \( (a, b) \) is the interval on which \( u(x) \) has non-zero membership values, \( \lambda \) is the sharpness parameter, and \( v \) is interpreted as an expectation level. It is easy
to show that this membership function has the form of $1/(1 + d(x))$, when the distance function is

$$d(x) = \left( \frac{v}{1 - v} \right)^{1-\lambda} \left( \frac{b - x}{x - a} \right)^{\lambda}.$$

All methods reviewed in this category lack a general principle (such as maximum likelihood to estimate probability density). However, it should be noted that our understanding of how the human mind perceives and manipulates vague categories is quite primitive and inconclusive.

3. Heuristic methods

Heuristic methods use predefined shapes for membership functions and have been used successfully in rule-based pattern recognition applications [9]. In computer vision, heuristic membership functions may be used to describe certain spatial relations (such as “above” and “to the left of” [10,11]) and certain properties (such as lightness or darkness of a pixel value, position of a pixel, and narrowness of a region). Here, we present a few frequently used shapes for heuristic membership functions.

*Piecewise linear functions:*

The membership functions may be chosen to be linearly increasing, linearly decreasing or a combination of these.

**Example 1.**

$$\mu(x) = 1 - \frac{x}{a}; \quad \mu(x) = \frac{x}{a}; \quad \text{where } X = [0, a] \text{ is a reference set.}$$

**Example 2.**

$$\mu(x) = \begin{cases} 
1 - \frac{|a - x|}{a} & \text{if } a - a \leq x \leq x + a \\
0 & \text{otherwise.}
\end{cases}$$

The membership functions may also be chosen as piecewise linear functions, i.e., they have linearly increasing, decreasing and flat regions.

**Example 3.**

$$\mu(x) = \begin{cases} 
0 & \text{if } x \leq a \\
x - a & \text{if } a \leq x \leq b \\
w_1 \frac{x - a}{b - a} & \text{if } a \leq x \leq b \\
1 & \text{if } b \leq x \leq c \\
w_2 \frac{d - x}{d - c} & \text{if } c \leq x \leq d \\
0 & \text{if } x > d.
\end{cases}$$
Example 4.

$$\mu(x) = \begin{cases} 
0 & \text{if } x < a_1 \\
ax + b & \text{if } a_1 \leq x \leq a_2, \\
1 & \text{if } x > a_2 
\end{cases}$$

where $$a = \frac{1}{a_2 - a_1}$$ and $$b = \frac{a_1}{a_1 - a_2}$$.

**Piecewise monotonic functions:**

In this case, membership functions have a (piecewise) smooth transition between non-membership and full-membership regions. The smooth transition may be described by functions such as $$x^2$$, $$\sin(x)$$, $$\arctan(x)$$, and $$\exp(x)$$.

Example 5. The $S$-function

$$S(x; a, b, c) = \begin{cases} 
0 & x \leq a \\
2\left(\frac{x - a}{c - a}\right)^2 & a < x \leq b \\
1 - 2\left(\frac{x - a}{c - a}\right)^2 & b < x \leq c \\
1 & x > c 
\end{cases}$$

with $$b = \frac{a + c}{2}$$.

Example 6. The $\pi$-function

$$\Pi(x; a, b, c) = \begin{cases} 
S\left(x; c - b, c - \frac{b}{2}, c\right) & x \leq c \\
1 - S\left(x; c, c + \frac{b}{2}, c + b\right) & x > c. 
\end{cases}$$

Example 7.

$$\mu(x) = \exp^{-b(x-a)^2}.$$  

Example 8.

$$\mu(x) = \frac{1}{2} - \frac{1}{2} \sin \left( \frac{\pi}{b-a} \left( x - \frac{a+b}{2} \right) \right), \quad x \in [a, b].$$

Dombi’s review of heuristic functions [8] found that the membership functions had the following common features: (i) all membership functions are
continuous, (ii) all membership functions map an interval \([a, b]\) to \([0, 1]\), (iii) membership functions are either monotonically increasing or monotonically decreasing or both increasing and decreasing.

Since heuristic methods are chosen to fit the given problem, they work well only for problems for which they are intended. Linear and piecewise linear membership functions have the following advantages: they provide a reasonably smooth transition, they are easily manipulated by fuzzy operators, and they are easily implemented in hardware if speed is crucial.

Unfortunately, the shapes of the heuristic membership functions are not flexible enough to model all kinds of data. Moreover, the parameters associated with the membership functions must be provided by experts. In some applications, the parameters need to be "tweaked" until the performance is acceptable. This tuning process is non-trivial in a high-dimensional system due to interactions between variables and local minima.

4. Histogram based methods

Histograms of features provide information regarding the distribution of input feature values. A multidimensional histogram of \(n\)-dimensional feature vectors from regions of interest in an image can be constructed for each class/region. The histogram thus generated can be modeled by a mixture of parameterized functions such as Gaussians. The parameterized mixture can then be used as the membership function for the particular class/region. This method is easy to implement, and memberships once generated can be used for classification in the testing phase.

Fig. 1(a) shows the intensity image of a 256 \(\times\) 256 color image obtained from University of Massachussets. The excess-green \((2g - r - b)\) and intensity \((r + g + b)/3\) components of the color image were used as the input features. Regions corresponding to each of the three classes (sky, vegetation and road) were specified by the user. In this example, a single Gaussian was used to model the data for each class, rather than a mixture. The feature values at each pixel in the user-specified rectangular windows within each region were used to find the mean vector and covariance matrix for the 3 bivariate Gaussians representing the three classes. The sizes of the windows used for the three classes were (a) vegetation: 50 \(\times\) 60 and, (b) sky: 40 \(\times\) 50, and (c) road: 45 \(\times\) 60. The contours of the 3 bivariate Gaussians found after the modeling process, mapped onto the feature space are shown in Fig. 1(b). These membership functions are then used to classify the entire feature space. The result of classification (based on highest membership) is shown in Fig. 1(c). The membership values for every pixel in the image in each of the three classes are shown in Fig. 2(a)–(c).
Fig. 1. (a) Original outdoor scene image. (b) Contours of the membership functions mapped onto Feature space. (c) The corresponding segmented result.

Fig. 2. Membership values of pixels in Fig. 1(a) in each of the three Classes after training: (a) Class "Tree"; (b) Class "Sky"; (c) Class "Road".
5. Transformation of probability distributions to possibility distributions

Possibility theory, originally proposed by Zadeh [12], stems from the fact that most of the information on which human decisions are based is possibilistic rather than probabilistic in nature. One of the central concepts of possibility theory is that of a possibility distribution. The problem of converting probability distributions to possibility distributions has been studied by many authors. The conversion problem has its root in the possibility/probability consistency principle defined by Zadeh [12]. More recently Dubois and Prade [13,14], and Klir [15] have investigated several kinds of scales for the transformation.

The possibility/probability consistency principle [12] states:

\[
C_z(\pi, p) = \sum_{i=1}^{n} \pi(u_i) p(u_i).
\]

An alternative definition of consistency [12] is as follows:

\[
C_{DP}(\pi, p) = \begin{cases} 
1 & \text{if } P(A) \leq \Pi(A) \forall A, A \subset U \\
0 & \text{otherwise.}
\end{cases}
\]

Here \( U \) is the universal set, \( P(A) \) denotes the probability of \( A \), and \( \Pi(A) \) denotes the possibility of \( A \), i.e., \( P(A) = \sum_{u \in A} P(u) \) and, \( \Pi(A) = \max_{u \in A} \{ \pi(u) \} \).

However, Zadeh [12] affirms that the possibility/probability consistency principle is not a precise law or a relationship that is intrinsic in the concepts of possibility and probability. Rather it is an approximate formalization of the heuristic observation that a lessening of the possibility of an event tends to lessen its probability but not vice versa. The possibility/probability consistency principle provides a basis for the computation of the possibility distribution corresponding to the probability distribution of \( X \).

In many applications, normalized histograms have been traditionally treated as probability distributions. If we have a large number of samples, the normalized histogram of the samples can be assumed to approximate the pdf. If we treat membership functions as numerically equivalent to possibility distributions [12], then methods that transform probabilities to possibilities can be used to generate membership functions from histograms. We discuss two such transformation methods: the Bijective Transformation method by Dubois and Prade [14] and the Conservation of uncertainty method by Klir [15].
5.1. Bijective transformation method

Let \( X = \{ x_i \mid i = 1, \ldots, n \} \) be the universe of discourse. The \( x_i \)'s are ordered such that \( p_1 \geq p_2 \geq \cdots \geq p_n \), where \( p_i \) is the probability of occurrence of \( x_i \), i.e., \( p_i = P(\{x_i\}) \). Let \( \pi_i \) denote the corresponding possibility value. A bijective transformation between probabilities and possibilities may be defined as [14]:

\[
\pi_i = \sum_{j=1}^{n} \min(p_i, p_j) = ip_i + \sum_{j=i+1}^{n} p_j \tag{2}
\]

and

\[
p_i = \sum_{j=1}^{n} \frac{\pi_j - \pi_{j+1}}{j}
\]

with the convention \( \pi_{n+1} = 0 \). This mapping was derived from the definition that the degree of necessity of event \( A \) in \( X \) is the extra amount of probability of elementary events in \( A \) over the amount of probability assigned to the most frequent elementary event outside \( A \). From Eq. (2), it is seen that the overall shape of the possibility distribution is the same as that of the probability distribution, and vice versa, i.e.,

\[
\pi_i = \pi_{i+1} \iff p_i = p_{i+1}, \quad \pi_i > \pi_{i+1} \iff p_i > p_{i+1}.
\]

**Theorem 1.** The possibility distribution represented by \( \pi_i \) is greater or equal to the corresponding normalized probability distribution. That is, \( \pi_i \geq p_i/p_{\max} \forall \ i \in \{1, 2, \ldots, n\} \), where \( p_{\max} = \max_i(p_i) \).

**Proof.** Multiplying both sides of Eq. (2) by \( p_i/p_{i+1} \),

\[
\frac{p_{i+1}}{p_i} \pi_i = ip_i + \frac{p_{i+1}}{p_i} \sum_{j=i+1}^{n} p_j \geq \sum_{j=1}^{i} p_j + \sum_{j=i+1}^{n} p_j = \sum_{j=1}^{n} p_j = 1
\]

since \( p_i \geq p_j \) for all \( i \). Hence, \( \pi_i \geq \pi/p_{i+1} \).

5.2. Conservation of uncertainty method

A similar approach was taken by Klir [15] based on the principle of uncertainty conservation. He suggested that, when uncertainty is transformed from one theory \( T_1 \) to another \( T_2 \) the following requirements must be met:

1. The amount of inherent uncertainty should be preserved when the transformation is made from \( T_1 \) to \( T_2 \).
2. All relevant numerical values in \( T_1 \) must be converted to their counterparts in \( T_2 \) by an appropriate scale.

The probabilistic measure of uncertainty is the well known Shannon entropy and is given by
In possibility theory, there are two types of uncertainties, non-specificity $N(\pi)$, and discord $D(\pi)$, and they are given by

$$N(\pi) = -\sum_{i=1}^{n} \pi_i \log_2 \left( \frac{i}{i-1} \right)$$

and

$$D(\pi) = -\sum_{i=1}^{n-1} (\pi_i - \pi_{i+1}) \log_2 \left[ 1 - i \sum_{j=i+1}^{n} \frac{\pi_i}{j(j-1)} \right].$$

Therefore, the principle of uncertainty conservation can be expressed as

$$H(p) = N(\pi) + D(\pi).$$

Klir [15] contends that the log-interval scale transformation is the only one that exists for all distributions and is unique. Its form is

$$\pi_i = \left[ \frac{P_i}{P_1} \right]^{\alpha},$$

where $\alpha$ is a positive constant determined by solving (3); Klir conjectures that $\alpha$ lies in the interval $[0, 1]$.

Pal et al. [16,17] discussed the limitations of the measures of total uncertainty. Their analysis reveals that total uncertainties have several maxima which makes it difficult to gauge the quality of evidence based on numerical values. They also indicate that elementary measures of probabilistic uncertainty or non-specificity such as dissonance and discord measure only one or two aspects of uncertainty making the interpretation of total uncertainty difficult.

The transform based methods are intended for situations where the manipulation of randomness is hard, and where it is easier to handle uncertainty via possibility distributions in the fuzzy framework. The transform methods have the disadvantage of requiring a large amount of data to estimate the probability density function. On the other hand, the general shapes of the probability functions and possibility functions are the same. Therefore, normalized histograms can be used as membership functions directly (see Section 4). This approach assumes that membership degree is the same as the frequency of occurrence. However, it should be admitted that randomness represented by probability theory and vagueness represented by fuzzy set theory are inherently different concepts [18]. Therefore, the validation of this interpretation of memberships remains in question.

We now present examples of probability to possibility transformation. We used a synthetic 2-D Gaussian distribution (mean = (15, 15), $\sigma = 5$) (shown in Fig. 3(a)) to represent the probability distribution of a hypothetical data set.
Fig. 3. Probability to possibility transformation: Contour plots of (a) the original probability distribution; (b) the resulting possibility distribution using the Dubois-Prade technique; and (c) the resulting possibility distribution using Klir's technique.
The transformation between probability to possibility distribution can be done using either the Dubois–Prade technique or Klir's method. We divided the 2-dimensional probability distribution into bins, and computed the frequencies in each of the bins. The frequencies were then sorted in descending order, and using Eq. (2) and Eq. (3) the transformation was made to the 2-D possibility domain. We experimented with several bin sizes and found that we obtained the best results for bin size equal to 33. The result using the Dubois–Prade technique is shown in Fig. 3(b). Klir's technique yields the result shown in Fig. 3(c). Clearly, the possibility distribution is an upper envelope of the probability distribution.

6. Fuzzy nearest neighbor techniques

Let \( X = \{x_1, x_2, \ldots, x_n\} \) be set of \( n \) labeled samples, and let \( x'_n \in X \) be the sample nearest to \( x \). The nearest neighbor rule for classifying \( x \) is to assign it the label associated with \( x'_n \). The nearest neighbor rule is a sub-optimal procedure, the use of which, in the worst case, can result in an error that is twice the minimum possible Bayes rate [19]. A simple extension of the nearest neighbor rule is the \( K \)-nearest neighbor rule which classifies a sample vector \( x \) by assigning it the label that is represented by a majority of the \( K \)-nearest samples. The simplicity of implementation and the good results obtained have made \( K \)-nearest neighbors (\( K \)-NN) algorithm very popular. The theory of fuzzy sets was introduced to the nearest neighbor domain by Keller et al. [20]. The Fuzzy \( K \)-NN algorithm assigns class memberships to a sample vector rather than assigning the vector to a particular class. This ensures that no arbitrary assignments are made. The memberships assigned to the sample vector are dependent on the sample's distance from its \( K \)-NN and memberships of those \( K \)-NN in the possible classes.

Let \( X = \{x_1, x_2, \ldots, x_n\} \) be the set of labeled samples. Let us also assume that \( u_i(x) \) is the membership of vector \( x \) (to be computed) in class \( i \). Then, according to the fuzzy \( K \)-NN rule

\[
u_i(x) = \frac{\sum_{j=1}^{K} u_{ij} \left( \frac{1}{d_j^2} \right)^{1/(m-1)}}{\sum_{j=1}^{K} \left( \frac{1}{d_j^2} \right)^{1/(m-1)}},
\]

where \( u_{ij} \) is the membership of the \( j \)th labeled vector in the \( i \)th class, and \( d_j \) is the distance between feature vector \( x \) and the \( j \)th labeled vector. The assigned membership values are influenced by the inverse distances of the nearest neighbors and their class memberships. The labeled samples (training data) can be assigned class memberships in several ways [20]. However, for our experiments, we assigned memberships to our labeled data points using the fuzzy 5-nearest neighbor rule. In this technique, the \( K = 5 \) nearest neighbors to each
Generation of membership functions using the fuzzy K-nearest neighbors technique is illustrated using the Iris data set. The Iris data set consists of four input features, 150 patterns and three classes (50 in each class). Since Features 3 and 4 of the Iris data set contain more discriminatory information, only Features 3 and 4 were used in this experiment. In Fig. 4, we show the scatter plot of Feature 3 vs. Feature 4 of the Iris data set. We used 40 labeled samples from each class as the training data set. The fuzzy 5-nearest neighbor algorithm was used for training (i.e., for assigning fuzzy labels to training data), and the remaining 10 samples in each class were used for testing. The classification results obtained for different values of $K$ are presented in Table 1.

In another experiment, all of the 50 labeled samples per class were used as the training data set, and the entire feature space (uniformly sampled) was used as the testing set. The resulting membership functions (for $K = 5$) are shown in Fig. 5(a)–(c).
Table 1
Number of misclassified vectors using the $K$-nearest neighbor Classifier for different values of $K$

<table>
<thead>
<tr>
<th>$K$</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

The primary use of the nearest neighbor techniques involves situations where the a priori probabilities and class conditional densities are unknown. The algorithm does not generate smooth membership curves in overlapping regions as seen in Fig. 5(c). However, this algorithm is one of the simplest among the algorithms discussed in this paper.

7. Neural network based methods

Feedforward multilayer neural networks can be used to generate membership functions [18,21,22] from labeled training data. The output values of a sigmoid activation function of a neuron are quite similar to the membership values of fuzzy sets. The number of input nodes in the neural network is chosen to be equal to the number of features, and the number of output nodes is chosen to be equal to the number of class labels. The desired value of the output for an input feature vector is 1 for the node representing the label associated with the feature vector and 0 for all the other output nodes. In order to generate class membership values, a multilayer network is trained using a suitable training algorithm such as the back-propagation algorithm. After the training procedure converges, the resulting network can be treated as a membership generation network, where the inputs are feature values and the outputs are membership values in the different classes. This approach can be justified from an empirical point of view since activation functions have the same form as the membership function proposed by Zimmerman and Zysno [7] and Kochen and Badre [6] (see Section 2). In a neural network scenario, $a$ and $b$ in Eq. (1) can be interpreted as the weight and bias respectively.

This method allows fairly complex membership functions to be generated because the network is highly nonlinear in general. Since the membership functions are generated from a classification point of view, they are highly suitable for pattern recognition applications, although the membership values may not be necessarily indicative of the degree of typicality of a feature with respect to a class. One disadvantage of this method is that the shape of the membership function is unpredictable in regions where there is no training data. This problem can be overcome by introducing artificial data with all-zero targets (desired output values). Since the regions where there are no training
Fig. 5. Membership functions generated for the Iris data set using the K-NN technique. (a) Class A; (b) Class B; and (c) Class C.
Fig. 6. Membership functions generated by the neural network method for the three classes in the Iris data set.
data are usually far away from the ideal members, it is reasonable to assign all-zero targets to these artificial points.

We now illustrate this method using the Iris data set (see Fig. 4). The neural network was initialized with 2 input units (for Features 3 and 4), 6 hidden units, and three output units (representing the three classes). The network was trained using the back-propagation algorithm. The 150 input patterns were used in the training set, for the testing set, a quantized feature space (Features 3 and 4) was used. The resultant values at the output nodes correspond to the memberships in the appropriate classes. The membership curves of the three classes are presented in Fig. 6(a)–(c), for membership values of 0.2, 0.4, 0.6, 0.8 and 0.9. From Fig. 6, we can see that within a class, there is not much gradation in membership values. Thus, non-typical and typical elements of a class have similar membership values. Also, the shape of the membership function is unpredictable in regions where no training data exists. To overcome this problem, we introduce artificial data points with zero targets. Fig. 7 shows a scatter plot of the modified Iris data set with 150 randomly selected zero-target points. The resulting membership function contours are shown for the three classes in Fig. 8(a)–(c). The membership contours represent membership values equal to 0.2, 0.4, 0.6, 0.8 and 0.9. The membership functions in this case have zero values in regions where no original data patterns exist.

![Fig. 7. Scatter plot of modified Iris data set (Feature 3 vs. Feature 4). Crosses represent added zero target patterns, and “squares” represent the original Iris data patterns.](image-url)
Fig. 8. Membership functions generated by the neural network method for the three Classes in the modified Iris data set; (a) Class A; (b) Class B; and (c) Class C.

8. Methods based on clustering

8.1. The Fuzzy C-Means method

The Fuzzy C-Means (FCM) [23] algorithm is one of the most popular fuzzy clustering algorithms. Let \( X = \{x_1, x_2, \ldots, x_n\} \) denote a data set. Let \( n \) be the number of data vectors and \( c \) be the number of classes. The FCM algorithm does the partitioning by minimizing the objective function

\[
J(U, V) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m d_{ik}^2 \quad \text{subject to} \quad \sum_{i=1}^{c} u_{ik} = 1, \tag{4}
\]

where \( u_{ik} \) is the membership of \( x_k \) in cluster \( i \), \( d_{ik} \) is the distance from sample \( x_k \) to cluster prototype \( v_i \), \( V \) is the collection of all prototypes \( v_i \), and \( m \in [1, \infty) \) is a weighting exponent called the fuzzifier. The matrix \( U = [u_{ik}] \) represents the
fuzzy partition generated by the algorithm. It has been shown that for \( m > 1 \), under the assumption that \( x_k \) is not equal to \( v_i \) for all \( i, k \), \((U, V)\) may be a global minimum for \( J_m \) only if

\[
u_{ik} = 1 \left/ \sum_{j=1}^{c} \left( \frac{d_{ik}^2}{d_{jk}^2} \right)^{1/(m-1)} \right.
\]

and

\[v_i = \sum_{k=1}^{n} (u_{ik})^m x_k / \sum_{k=1}^{n} (u_{ik})^m.
\]

Starting with arbitrary prototype vectors \( v_i \), Eqs. (5) and (6) are used in an alternating fashion to generate a fuzzy partition of the data. The advantages of the FCM algorithm are: (i) it can be used as an unsupervised algorithm, (ii) it can be used to generate multi-dimensional membership functions, and (iii) the shape of the membership functions can be controlled by using different types of distance measures. However, the number of classes must be provided to run the algorithm. Owing to the constraint in Eq. (4), the membership values generated by the FCM algorithm do not represent degrees of belonging (or typicality), but rather "degrees of sharing". Moreover, the memberships cannot distinguish between a moderate outlier and an extreme outlier [24,25]. This makes the FCM algorithm sensitive to outliers.

When the data is labeled, it does not make sense to use the FCM algorithm in the above mode. Instead, we should perform "supervised" clustering on each class separately. We now present an example which uses the FCM algorithm to generate membership functions in the supervised mode. To generate membership functions using the FCM algorithm in the supervised mode, each of the classes of the Iris data set were separately clustered to find multiple prototypes per class. The estimated prototypes were then used along with an assumed distance measure to generate the membership value of a test pattern \( x_j \) in each class. Let \( d_{\text{min}}^2 \) denote the distance between \( x_j \) and the closest prototype in class \( k \) for \( k = 1, \ldots, c \). Let the prototypes in class \( k \) be denoted by \( v_1^k \ldots v_{n_k}^k \), where \( n_k \) is the number of prototypes used for class \( k \). Then,

\[d_{\text{min}}^2 = \min_p ||x_j - v_p^k||^2, \quad p \in \{1, \ldots, n_k\}.
\]

The membership values for \( x_j \) in class \( k \) is then computed using

\[u_{kj} = \left( d_{\text{min}}^2 \right)^{-1} / \sum_{i=1}^{c} \left( d_{\text{min}}^2 \right)^{-1}.
\]

The optimum number \( n_k \) of prototypes in class \( k \), for \( k = 1, \ldots, c \), was determined by trying various combinations. The combination \((n_1, \ldots, n_c)\) which
produced the least error on the training data is presented here. The best combination for the Iris data set consisted of 1 cluster in Class A, 2 clusters in Class B and 3 clusters in Class C, i.e., \((n_1, \ldots, n_c) = (1, 2, 3)\).

We used Features 3 and 4 and Euclidean distance to cluster each of the classes (40 training vectors per class). The membership functions generated as a result of this supervised clustering were then used to classify the entire feature space of Features 3 and 4. The membership contours corresponding to 0.5, 0.7, 0.8, 0.9 levels are shown for each of the classes in Fig. 9(a), (c) and (e) respectively. This technique clearly shows how membership functions can be generated when there are multiple clusters in each class. The training data points for each of the classes are also shown as crosses in Fig. 9(a), (c) and (e). These membership functions were also used to test the remaining 10 points from each class that were not used in the training process. The membership functions gave rise to a total of eight mis-classifications.

As mentioned above, traditional clustering methods are sensitive to noise. To illustrate the effect of noise, we modified the Iris data set by adding impulse and Gaussian noise to each element of the feature vectors in the data set. A scatter plot of the noisy version of the Iris data set is shown in Fig. 10. The above membership generation procedure was then used to find the membership functions for the classes in the noisy version of Iris data. The resulting membership functions for the modified Iris data set are given in Fig. 9(b), (d) and (f). Clearly, the shapes of the membership functions have been affected due to the presence of noise.

8.2. Robust agglomerative gaussian mixture decomposition (RAGMD)

Here, we present an overview of a recently introduced Robust Agglomerative Gaussian Mixture Decomposition (RAGMD) algorithm [26] that can model data by automatically determining the correct number of components in the presence of noise. The RAGMD algorithm is an extension of the EM algorithm [27] and achieves robustness by using the least trimmed squares approach [28]. The objective function of the RAGMD is

\[
J_{RA} = \sum_{j=1}^{P} \log \left( p(x_{j:n} | \theta) \right) + z \sum_{i=1}^{c} P(\omega_i) \log(P(\omega_i)),
\]

where

\[
p(x_j | \theta) = \sum_{i=1}^{c} P(\omega_i) p(x_j | \omega_i, \theta_i) \quad \text{and} \quad p(x_j | \omega_i, \theta_i) = N(m_i, C_i).
\]

In Eq. (8), \(x_{1:n}, \ldots, x_{n:n}\) represent an ordered version of \(x_1, \ldots, x_n\) such that \(p(x_{1:n} | \theta) \geq \cdots \geq p(x_{n:n} | \theta)\), where \(P\) is the number of data vectors used in the minimization, \(n\) is the total number of vectors, \(c\) is the number of components,
and $\theta = (\theta_1, \ldots, \theta_c)$ represents the mixture density parameters for the $c$ components. $\theta_i = (\mu_i, C_i, P(\omega_i))$, where $\mu_i$ is the mean, $C_i$ the covariance matrix, and $P(\omega_i)$ the a priori probability of component $i$. When $\theta$ is correctly esti-
mated, the value of \( p(x_j|\theta) \) is high for good points and low for bad points. Hence \( p(x_j|\theta) \) can be used as a measure of typicality of \( x_j \) in the class of good points.

The objective function in Eq. (8) cannot be maximized easily, and the update equations are obtained by a heuristically extending the update equations of the EM algorithm [29]. The advantage of this robust method is that only the retention ratio \( P \) needs to be specified and in most applications a rough figure for the degree of noise is known.

We now present an example in which the RAGMD algorithm was used in the "supervised" mode on the Iris data set to generate membership functions for each of the classes separately. Supervised clustering was performed on 40 points from each class using Features 3 and 4 to extract the prototypes \( \theta_i \) representing the classes. The prototypes were used to generate memberships using the following equation

\[
    u_{ij} = p(x_j|\beta_i) = \sum_{k=1}^{n} P(\omega_k)p(x_j|\omega_k),
\]

where \( u_{ij} \) is the membership of \( x_j \) in class \( \beta_i \), \( n_i \) is the number of components being used to model class \( \beta_i \), \( P(\omega_k) \) is the mixing parameter and \( p(x_j|\omega_k) \) is the conditional density function corresponding to the \( k \)th component in class \( \beta_i \). The resulting membership functions were used to classify the remaining \( 10 \times 3 = 30 \) points. This robust clustering technique resulted in a total of 4 misclassifications. The membership functions for the 3 classes in the Iris data set are shown in Fig. 11(a), (c) and (e) respectively. The membership functions obtained using the RAGMD algorithm are more compact compared to those
Fig. 11. Membership functions generated by using the RAGMD method. The membership function contours at levels 0.01, 0.5 and 0.9 for Iris data: (a) Class A, (c) Class B and (e) Class C. The membership function contours at levels 0.01, 0.5 and 0.9 for the noisy version of the Iris data; (b) Class A; (d) Class B; and (f) Class C.
from the FCM algorithm. Since RAGMD is a robust algorithm, we tested it on the modified Iris data set shown in Fig. 10. The resulting membership function contours for the 3 classes are shown in Fig. 11(b), (d) and (f) respectively. It can be seen that the overall shape of the membership function roughly remains the same.

9. Conclusions

In this paper, we investigated several memberships generation methods for pattern recognition. We reiterate that there is no single best method, and the choice of the method depends on the particular problem. The applicability of heuristic methods is limited because they are not sufficiently flexible and they cannot use training data, especially in high dimensions. Multi-dimensional histograms can be used to generate membership functions, however, this approach needs a lot of data to generate smooth membership functions. The transformation based methods work well in situations where randomness is hard to manipulate and where possibility distributions are easy to handle. The fuzzy $K$-nearest neighbors technique can be used when the a priori probabilities and class conditional densities are unknown, and it is easy to implement. In theory, neural nets are capable of producing complex membership functions. However, typical and non-typical elements of a class have similar membership values. By using fuzzy targets we can alleviate the problem of non-graded memberships within a class generated by the neural network method. Supervised clustering techniques are suitable for pattern recognition applications which involve training data. The shape of the membership functions generated by fuzzy and robust clustering algorithms can be controlled by the type of distance measure and other parameters used. The FCM memberships are acceptable in classification applications involving no noise. On the other hand, the RAGMD algorithm is more general and can be used when data needs to be modeled using mixtures.

Acknowledgements

Partial support of this work by a grant from the office of Naval Research (N00014-96-1-0439) is gratefully acknowledged.

References


