Formulation of a Multivalued Recognition System

Deba Prasad Mandal, C. A. Murthy, and Sankar K. Pal, *Senior Member, IEEE*

***Abstract*—A recognition system based on fuzzy set theory and approximate reasoning has been described that is capable of handling various imprecise input patterns and providing a natural decision system. The input feature is considered to be of either *quantative form* or *linguistic form* or *mixed form* or *set form*. The entire feature space is decomposed here into some overlapping *subdomains* depending on the geometric structure and the relative position of the pattern classes found in the training samples. The various uncertainty (ambiguity) in the input statement has been managed by providing/modifying membership values to a great extent. A relational matrix corresponding to the *subdomains* and the pattern classes has been considered in the modified Zadeh's compositional rule of inference in order to recognize the samples. The linguistic output decision is associated with a confidence factor denoting the degree of certainty of a decision. The effectiveness of the algorithm has been demonstrated on some artificially generated patterns and also on the real life speech data. The recognition scores are described in terms of various choices namely, *single correct*, *first correct*, *combined correct*, *second correct* and *fully wrong choices*; thus provides a low rate of misclassification as compared to the conventional two-state systems.**

I. Introduction

THE PATTERN CLASSIFICATION methods can primarily be grouped into two categories; namely decision theoretic [1], [2] and syntactic [3]. In these conventional classifiers, the input patterns are quantitative (exact) in nature. They provide crisp (two-state) output and are mostly suitable for the mechanistic type of problems. The patterns having imprecise or incomplete information are usually ignored or discarded from their designing and/or testing processes. The impreciseness (or ambiguity) [4], [5] may arise from various reasons. For example, instrumental error or noise corruption in the experiment may lead to have partial (incomplete) information available on a feature measurement F viz., F is about 500 or F is between 400 and 500 etc. Again, in some cases the expense incurred in extracting exact value of feature may be high or it may be difficult to decide on the actual salient features to be extracted. On the other hand, it may become convenient to use linguistic variables or hedges e.g., small, medium, high, more or less, very, etc. in order to describe feature information.

A decision theoretic recognition system based on fuzzy set theory [6], [7] and approximate reasoning [8]–[11] is designed to be capable of handling all the aforesaid impreciseness. Initially, each individual feature range is divided into some domains depending on the geometric complexity and the relative positions of the pattern classes found in the training samples. To handle the impreciseness of the input feature information and to incorporate the portions possibly uncovered by the training samples, each of the domains is extended to some extent using triangular membership functions. As a result, the whole feature range is decomposed into few overlapping subdomains.

The theory of approximate reasoning [8] has been introduced by Zadeh in 1977. This theory has the capability to handle both soft and hard data as well as various types of uncertainty. Many aspects of the underlying concepts have been incorporated in designing decision making systems [11]–[17] along with their applications.

The proposed system uses Zadeh's compositional rule of inference [8] and gives a natural (linguistic) multivalued output decision associated with its certainty (or validity). The effectiveness of the algorithm has been demonstrated on some artificially generated pattern sets as well as speech recognition problem.

In Section II, a brief description of the recognition system is provided. The description of different blocks are provided in Sections III and IV. Results are discussed in Section V. Section VI finds the conclusion.

II. Multivalued Recognition System

A. Basic Concepts

The proposed recognition system is capable of handling various input patterns having feature information in *quantitative form*, *linguistic form*, *mixed form*, and *set form*. The system assumes that every pattern class is a union of nearly rectangular sets. Thus, initially the training sample set of every pattern class is divided into few groups of nearly rectangular shapes [18]. Further, depending on the relative positions of the sample groups in the feature space, the sample groups are again subdivided. Accordingly each individual feature space is then decomposed into some domains to highlight the obtained groups of training sets so that each feature information can be converted as the belongingness to the obtained domains to some degree. To handle the uncertainty of the input information and to incorporate the portions (of the pattern classes) possibly uncovered by the training samples, each of the feature domains is extended to some extent using triangular membership functions. Thus the whole feature space is divided into some overlapping subdomains. The aforementioned decomposition of the sample sets and the

Manuscript received March 7, 1991; revised November 1, 1992.

D. P. Mandal and C. A. Murthy are with the Electronics and Communication Sciences Unit, Indian Statistical Institute, 203 B. T. Road, Calcutta 700 035, India.

S. K. Pal is on leave from the Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta 700 035, India and is with the Software Technology Branch, Mail Code PT4, NASA Johnson Space Center, Houston, TX 77058.

dynamic ranges of the features constitute the preprocessing part of the recognition system. It is to be noted that the preprocessing is completely based on the training samples. Henceforth, the same notations and terminologies, as stated here, will be followed throughout the paper.

Notations and Terminologies

1) $F_1, F_2, \cdots, F_N$ denote the features, where N represents the number of features.
2) $C_1, C_2, \cdots, C_M$ denote the classes, where M represents the number of classes.
3) The variable i stands for the features, i.e., $i = 1, 2, \cdots, N$.
4) The variable j stands for the classes, i.e., $j = 1, 2, \cdots, M$.
5) $\hat{M}$ denotes the total number of training sample groups, that is
$$\hat{M} = \sum_{j=1}^{M} m_j$$
where m_j denotes the number of sample groups obtained from the training samples of class C_j through decomposition.
6) $\hat{N}$ denotes the number of subdomains in the whole feature space, i.e.,
$$\hat{N} = \prod_{i=1}^{N} n_i$$
where n_i denotes the number of domains in the ith feature space.
7) The feature regions in the individual feature spaces are referred as the *domains* and the regions in the whole feature space, which are the combinations of the domains in the individual feature space, are referred here as the *subdomains*. The variables g and h stand for the domains and subdomains respectively. The domains are denoted as D_{i1}, D_{i2}, $\cdots$, D_{in_i} $(i = 1, 2, \cdots, N)$ and the the subdomains are denoted as SD_1, SD_2, $\cdots$, $SD_{\hat{N}}$.
8) $CV(X) = (cv_1(X), cv_2(X), \cdots, cv_{\hat{N}}(X))$ represents a characteristic vector where the hth element $cv_h(X)$ denotes the degree of belonging of a feature information X to the hth subdomain.
9) R represents the relational matrix, which denotes the compatibility of various pattern classes corresponding to the subdomains. The order of R is $\hat{N} \times M$.
10) $S(X) = (s_1(X), s_2(X), \cdots, s_M(X))$ represents a class similarity vector where the jth element $s_j(X)$ denotes the degree of similarity of a pattern X to the jth class.

To explain the preprocessing concept, let us consider a 2 class and 2 feature problem (i.e., $M = 2$ and $N = 2$) as shown in Fig. 1(a). Based on the geometric structure [18], the sample set of class A is initially decomposed into two groups (denoted by A_1 and A_2) of nearly rectangular shapes as shown in Fig. 1(b). Then depending on the relative positions of the sample groups, the sample group A_1 is again subdivided into 2 subgroups A_{11} and A_{12} and the sample set of B is divided into two groups B_1 and B_2 (Fig. 1(c)). Hence there are five sample groups, i.e., $\hat{M} = 5$. Now in order to distinguish all the sample groups, the feature spaces F_1 and F_2 have been decomposed into 3 and 2 overlapping domains respectively. Thus there are 6 ($\hat{N} = 3 \times 2$) subdomains that highlight all the five sample groups. The subdomains with the reflected sample groups are shown in Fig. 1(d).

The relevance of the membership or compatibility functions for characterizing the subdomains is described in the next section.

B. Membership Functions

The preprocessing block of the recognition system decomposes each individual feature space into some overlapping domains. For a given pattern point in an individual feature space, the possibility of its being a member of a feature domain is maximum if it lies in the centre of the domain. As the distances of the points from the points in the central portion increase, the possibilities decrease and ultimately go to zero. All the triangular functions have the previous property. So any triangular function may be considered as the representative membership function for the domain of a feature space. As the π function (which is a quadratic triangular function) is well established to dictate the previous property [12], [19], it is considered here to serve the purpose.

Thus the gth domain along ith feature axis is characterized by $\pi_{ig}(x, \alpha_{ig}, \beta_{l_{ig}}, \beta_{u_{ig}}, \Gamma_{l_{ig}}, \Gamma_{u_{ig}})$ in which α_{ig} is the central part where the membership value is 1.0; $\beta_{l_{ig}}$ and $\beta_{u_{ig}}$ are the lower and upper most ambiguous (crossover) points where the membership values are 0.5; $\Gamma_{l_{ig}}$ and $\Gamma_{u_{ig}}$ are the lower and upper end points beyond which the membership values are zero. The functional form of such a π function is stated here:

$$\pi(x; \alpha, \beta_l, \beta_u, \Gamma S_l, \Gamma_u) = \begin{cases} S(x; \Gamma_l, \beta_l, \alpha) & \text{if } x \le \alpha \\ 1 - S(x; \alpha, \beta_u, \Gamma_u) & \text{if } x > \alpha \end{cases} \tag{1}$$

where

$$S(x; a, b, c) = \begin{cases} 0 & \text{if } x \le a \\ \frac{1}{2}\left(\frac{x-a}{b-a}\right)^2 & \text{if } a < x \le b \\ 1 - \frac{1}{2}\left(\frac{x-c}{b-c}\right)^2 & \text{if } b < x \le c \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

Such a π function is graphically shown in Fig. 2. Although the π function corresponding to a feature domain in a feature space ranges between Γ_l and Γ_u, it is assumed that only the portion between β_l and β_u of that feature space is represented by the training samples. The extented portions of the feature domain are $[\Gamma_l, \beta_l]$ and $[\beta_u, \Gamma_u]$. These extented portions take care of the possible uncovered regions by the training samples and the overlapping between different pattern classes.

Note that the π function is a quadratic function. For simplicity, a linear triangular function (T) may also be considered. The functional form of such a linear triangular function is

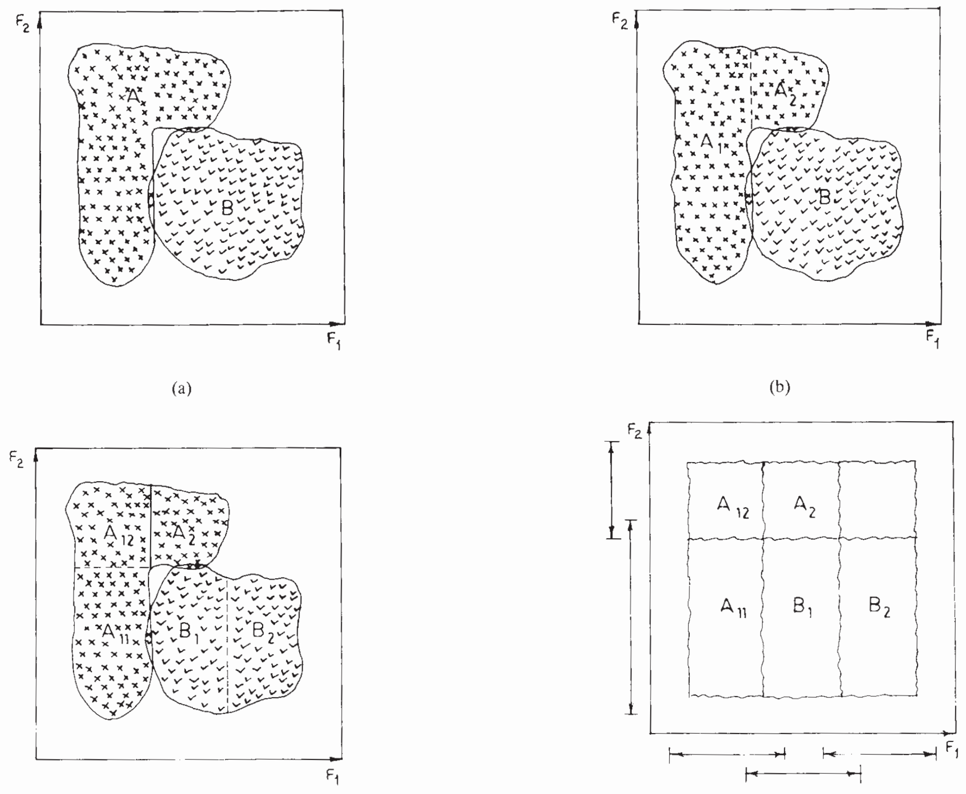


Fig. 1. (a)–(d) Showing the concept of preprocessing.

stated here:

$$T(x;\alpha,\beta_l,\beta_u,\Gamma_l,\Gamma_u)=\begin{cases}\frac{1}{2}\left(\frac{x-\Gamma_l}{\beta_l-\Gamma_l}\right) & \text{if } \Gamma_l \le x \le \beta_l\\ \frac{1}{2}+\frac{1}{2}\left(\frac{x-\beta_l}{\alpha-\beta_l}\right) & \text{if } \beta_l \le x \le \alpha\\ \frac{1}{2}+\frac{1}{2}\left(\frac{x-\beta_u}{\alpha-\beta_u}\right) & \text{if } \alpha \le x \le \beta_u\\ \frac{1}{2}\left(\frac{x-\Gamma_u}{\beta_u-\Gamma_u}\right) & \text{if } \beta_u \le x \le \Gamma_u\\ 0 & \text{otherwise.}\end{cases} \quad (3)$$

The structure of such a linear triangular function is shown in Fig. 3. The significants of all the parameters are exactly same with the previously stated π function. All the results in this paper are shown using the π membership functions for the domains in various feature spaces. Assuming the previous linear triangular function, similar results are obtained.

C. Block Diagram

The block diagram of the proposed recognition system is shown in Fig. 4. It consists of two sections, namely Learning and fuzzy processor. Learning section uses only the training sample information and finds the representative subdomains and a relational matrix. The fuzzy processor uses the relational matrix in the modified compositional rule of inference [8] to give a natural or linguistic output decision regarding the class or classes to which an unknown pattern X may belong.

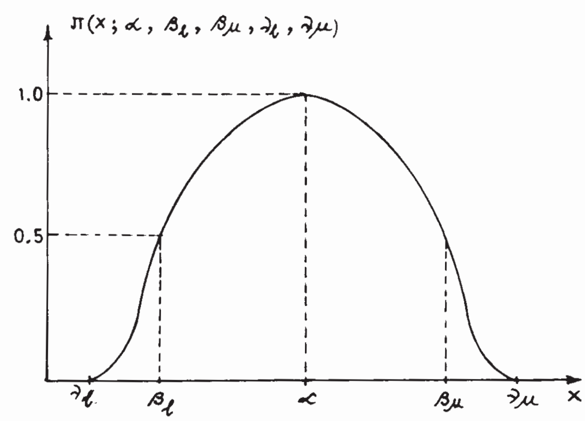


Fig. 2. π function.

The preprocessing task of the Learning section is explained previously. It decomposes the whole feature space into some overlapping subdomains. The relational matrix estimator block finds a relational matrix R.

The feature extractor block takes a pattern X as input and

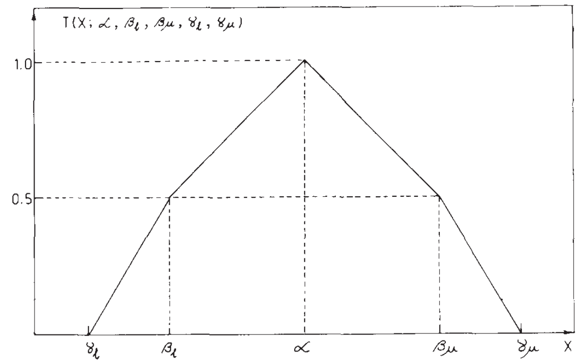


Fig. 3. Linear triangular function (T).

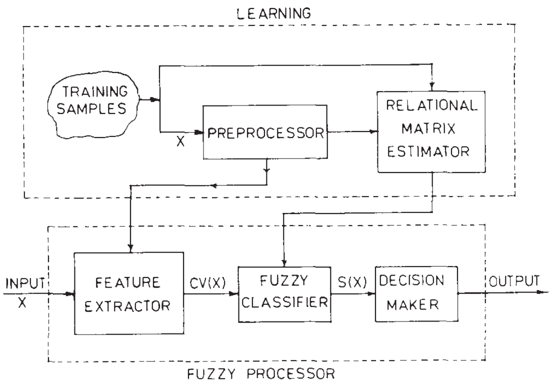


Fig. 4. Block diagram.

outputs a characteristic vector $CV(X)$. The fuzzy classifier block uses $CV(X)$ and R in the compositional rule of inference to find a class similarity vector $S(X)$.

Ambiguity (uncertainty) in the fuzzy decision, provided by $S(X)$, is then determined by computing CF (confidence or certainty factor). Higher the value of CF, the stronger is the validity of the decision. Depending on the value of CF, the final output of the recognition system is given in linguistic or natural form.

The recognition system described previously is referred here as multivalued because it normally gives multiple class choices with different degree of certainty of the classes. It may also be viewed as a generalized classifier providing natural (fuzzy and/or hard) output from both fuzzy and deterministic input.

The overview of the recognition system is provided previously. Various blocks of Fig. 4 are discussed in the following sections.

III. Learning

The operations of this section are fully dependent on the training samples. This section decomposes the whole feature space into some overlapping subdomains in order to handle the ambiguous information and estimates a relational matrix. It has two blocks, namely preprocessing and relational matrix estimator. The domains and subdomains in the feature space are obtained in the preprocessing block. The relational matrix estimator block finds a relational matrix R.

A. Preprocessing

In this block, geometric complexity [18] and the relative positions of the given pattern classes are considered one after another to decompose the training sample set of the pattern classes into some groups. Accordingly each individual feature space is divided into some domains to highlight the obtained sample groups. These concepts are explained in the following in two-dimensional (2-D) feature space.

Geometric Complexity: The system assumes that every pattern class is a union of nearly rectangular sets. In order to determine whether a pattern class is of nearly rectangular or not, an analysis based on overlapping windows is proposed. An accuracy factor (δ_T) based on the number of available training samples (say, T) is considered for deciding the rectangular property of the pattern classes. The value of δ_T is decided as [18]

$$\frac{1}{T^{0.49}} \le \delta_T \le \frac{1}{T^{0.33}} \tag{4}$$

so that as $T \to \infty$, $\delta_T \to 0$, and $T{\delta_T}^2 \to \infty$. Since δ_T decreases with the increase of T, the accuracy of the algorithm also increases with the increase of T. The inequality (4) is due to Grenander [20] who used it for estimation of set or class. The details are explained in [18].

The procedure is explained for a pattern class in the 2-D feature space. Here each class is considered separately. A typical training sample set is shown in a feature space in Fig. 5(a). To find the boundary variation of the set, four perpendicular directions (referred by the codes $1, 2, 3$ and 4), as shown in Fig. 5(b), are considered. Fig. 5(c) shows the boundaries of the sample set in the coded directions 1 and 2, where the first (F_1) and second (F_2) axes corresponds to the base and height respectively. Similarly, the Fig. 5(d) shows the boundaries of the sample set in the directions 3 and 4 (considering the F_2 and F_1 axes as the base and height respectively).

Formation of Windows: Let $(b_1, h_1), (b_2, h_2), \cdots, (b_T, h_T)$ be the sampled points in terms of base and height values. Initially from all the sampled points, the maximum (say, $\max_b$) and minimum (say, $\min_b$) of the base values are found. A base coverage factor, say ε_b, is defined as

$$\varepsilon_b = (\max_b - \min_b)\delta_T \tag{5}$$

where δ_T is the accuracy factor. All the windows are constructed from the training samples using ε_b so that the base coverage length of each window is atleast ε_b.

Similarly, the maximum (say, $\max_h$) and minimum (say, $\min_h$) of the height values are found and a height threshold factor, say ε_h, is defined as

$$\varepsilon_h = (\max_h - \min_h)\delta_T \tag{6}$$

where ε_h is used in deciding whether a sample set or group is of nearly rectangular or not.

Now the training samples are arranged in ascending order according to the base values. The first window starts with the

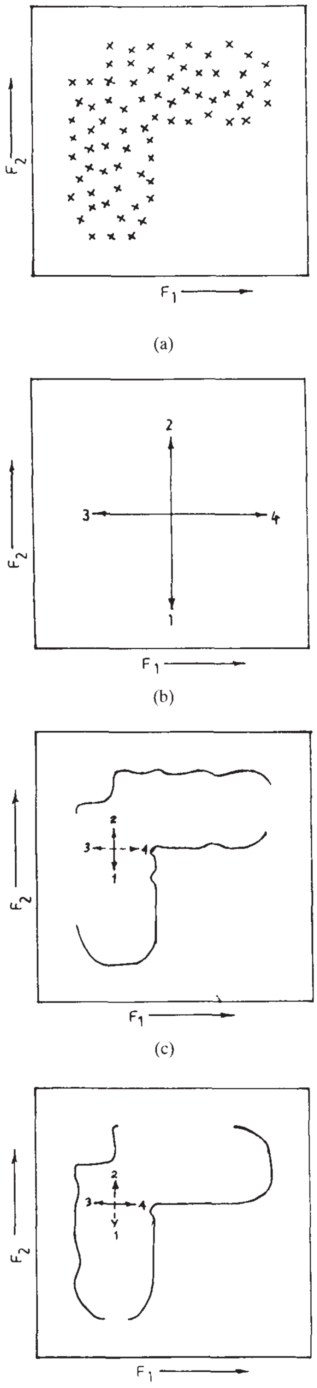


Fig. 5. (a)–(d) Showing the concept of obtaining boundary.

first sample of the ordered training samples and it includes all those samples one after another in ascending order until its base coverage length exceeds ε_b. Assume that the first window ends with the kth sample. Then the second window will end with the $(k+1)$th sample and to find the starting point of this window, it proceeds backward from kth sample until its base coverage length exceeds ε_b. Similarly, other windows are constructed. Thus, some overlapping windows of sample points are generated utilizing the sample base values and the base coverage factor ε_b.

The height values in a window are assumed to be the height coverages for that window area. The maximum and the minimum height sample values are chosen from each window to find the upper most and the lower most height coverage for a window, and these are considered here as the boundary values.

Calculation of Boundary Variations: To describe the procedure for calculating the boundary variations, let us consider the boundary in a particular direction, say d. The procedure to obtain the windows and the boundary values is discussed previously. Let us assume that there are w windows and their boundary values are H_k, $k = 1, 2, \cdots, w$. A boundary variation factor, say V_d, in the direction d is defined as

$$V_d = \left\{ \sum_{k=1}^{w-1} (H_k - H_{k+1})^2 \right\} \Big/ \varepsilon_h^2 \qquad (7)$$

where ε_h is the height threshold factor for the considered direction d.

Now let $\max_H$ and $\min_H$ be the maximum and minimum of the H_k's $(k = 1, 2, \cdots, w)$ respectively. If $(\max_H - \min_H) \leq \varepsilon_h$ then the variation factor V_d is assumed to be zero, i.e., make $V_d = 0$.

Initially, assuming F_1 and F_2 axes as the base and height respectively, the boundary variation factors V_1 and V_2 for the directions 1 and 2 are calculated. Similarly, by reversing the roles of F_1 and F_2 axes previously, the boundary variation factor V_3 and V_4 are obtained.

Pattern Class Subdivider: To determine the direction of decomposition, the algorithm finds the direction in which the variation is maximum. That is, the direction $D \in \{1, 2, 3, 4\}$ is obtained where $V_D \geq V_d$ for $d = 1, 2, 3, 4$. If $V_D = 0$, the sample set is assumed to be nearly rectangular in shape and it is not further decomposable.

Otherwise, i.e., if $V_D > 0$, it is assumed that the sample set is not nearly rectangular in shape and it is to be decomposed into groups. Now from the direction of decomposition (i.e., D) the windows with their base and boundary (or height) values and the corresponding height threshold value ε_h are recalled. The samples are arranged in ascending order according to the base values. For making a cluster of windows, the maximum boundary value is found. The starting window for the cluster is taken as the window where boundary value is maximum among all the boundary value. The position of the starting window is noted. The following windows for the starting window are arranged one after another in the cluster until the difference between the boundary values of the current window and the starting window is less than or equal to ε_h. Similarly, the preceding windows are also put in the window cluster. The samples lying in the previous window cluster are assigned to the first group of samples.

The previous routine is repeated on the remaining windows until all the windows are exhausted. This leads to the formation of window clusters. Every window cluster results in a group of sample points. Thus the given training sample set is decomposed into a few groups of sample points.

The decomposition procedure is applied on the sample groups repeatedly until all the groups are found to be nearly rectangular in shape.

Relative Position of Pattern Classes: The sample groups generated in the previous sections are nearly rectangular in shape. The relative positions of the sample sets in the feature space are then considered to divide (if necessary) further the sample sets so that the spans (ranges) of the sample sets in a group are more or less same in an individual feature space. This will facilitate (as described in the next section) to decompose the entire feature space into various subdomains. This concept of relative position has already been explained earlier by pattern diagrams in Fig. 1.

Here each feature axis is considered separately. Let us assume that there are $\hat{M}$ sample groups (initially $\hat{M} = M$). Let l_{ij} and u_{ij} be the lower and upper limits of the training samples corresponding to ith feature and jth sample group. Now follows an algorithm to decompose the training sample sets based on the relative positions of the sample groups along the first feature axis.

Algorithm I

Here two temporary sets of the sample sets namely covered set *cv-set* and lower set *l-set* are used.

Step 1: (Global initialization) *cv-set* =NULL; $\hat{M} = M$;

Step 2: $l\text{-}set = NULL$; $L_1 = \min_{\substack{j=1,2,\cdots,\hat{M}\\ j\notin cv\text{-}set}}\{l_{ij}\}$

$l\text{-}set = l\text{-}set + [j]$
if $l_{ij} = L_1$ for $j = 1, 2, \cdots, \hat{M}$ and $j \notin cv\text{-}set$:

$$U_1 = \min\left\{ \min_{\substack{j=1,2,\cdots,\hat{M}\\ j\notin cv\text{-}set\\ j\in l\text{-}set}} l_{ij}\ ,\ \min_{\substack{j=1,2,\cdots,\hat{M}\\ j\notin cv\text{-}set}} u_{ij} \right\}$$

$cv\text{-}set = cv\text{-}set + [j]$ if $u_{ij} = U_1$ for $j = 1, 2, \cdots, \hat{M}$ and $j \notin cv\text{-}set$;

Step 3: The sample groups belonging to the *l-set* and not to the *cv-set* are decomposed into two groups. In such cases, the training samples with first axis value less than or equal to U_1 are kept in the original sample group and include this sample group in the *cv-set*. Then a sample group is generated as $\hat{M} = \hat{M} + 1$ and the remaining samples are put in the new group.

Step 4: If *cv-set* includes all the sample groups, then the algorithm terminates. Otherwise go to Step 2.

An algorithm is described previously that decomposes the training sample set of a pattern class into groups according to the relative positions of the sample sets along the ith feature axis. Similarly the sample sets are decomposed depending on the relative positions of the sample sets along the other feature axes.

Decomposition of Feature Space: In order to highlight the generated sample groups, each individual feature space is divided into some overlapping domains. It is not difficult to group the sample sets so that the sample sets in each group correspond to one particular domain along an individual feature axis under consideration. The obtained domains are extended to some extent to incorporate the portions (of the pattern classes) possibly uncovered by the training samples and to handle the overlapping regions between the pattern classes. These domains are characterized by different π functions ((1)) of the form $\pi_{ig}(x, \alpha_{ig}, \beta_{l_{ig}}, \beta_{u_{ig}}, \Gamma_{l_{ig}}, \Gamma_{u_{ig}})$.

Each feature axis is considered separately. Let us assume that there are $\hat{M}$ sample groups that are obtained from the training samples of M pattern classes. Recall that the training samples of the jth class C_j is decomposed into m_j sample groups. Suppose l_{ij} and u_{ij} are the lower and upper most training samples corresponding to ith feature and jth sample group. Recall also that n_i denotes the number of domains in the ith feature space. An algorithm is described here to find domains (and hence the corresponding membership functions) along the ith ($i = 1, 2, \cdots, N$) feature axis.

Algorithm II

Here two temporary sets of the sample sets namely covered set *cv-set* and lower set *l-set* of sample groups are used. Let ext_{ig} be the extension factor decided based on the accuracy factor δ_T ((4)) for gth domain along the ith feature axis.

Step 1: (Global initialization) *cv-set* = NULL; $g = 0$;

Step 2: *l-set* =NULL; $g = g + 1$; $L_1 = \min_{\substack{j=1,2,\cdots,\hat{M}\\ j\notin cv\text{-}set}}\{l_{ij}\}$

$l\text{-}set = l\text{-}set + [j]$ if $l_{ij} = L_1$ for $j = 1, 2, \cdots, \hat{M}$ and $j \notin cv\text{-}set$; $U_1 = \max_{\substack{j=1,2,\cdots,\hat{M}\\ j\in l\text{-}set}}\{u_{ij}\}$ $cv\text{-}set = cv\text{-}set + l\text{-}set$;

Step 3: (Finding the parameters of π function) $ext_{ig} = (U_1 - L_1) \times \delta_T$; $\alpha_{ig} = (U_1 + L_1) \times 0.5$; $\beta_{l_{ig}} = L_1$; $\beta_{u_{ig}} = U_1$; $\Gamma_{l_{ig}} = L_1 - ext_{ig}$; $\Gamma_{u_{ig}} = U_1 + ext_{ig}$;

Step 4: If *cv-set* includes all the sample groups, then the algorithm terminates and assign $n_i = g$. Otherwise go to Step 2.

The previous algorithm decomposes the ith feature space into some (n_i) domains. This algorithm is repeated for all the feature axes. As a result, the total feature space is decomposed into few ($\hat{N} = \Pi_1^N n_i$) subdomains that incorporates all the sample groups. Note that $\hat{M}$ is not in general same with $\hat{N}$.

B. Relational Matrix Estimator

The relational matrix R denotes the compatibility of various pattern classes corresponding to the subdomains. The order of R is $\hat{N} \times M$, where $\hat{N}$ is the number of subdomains and M is the number of pattern classes. Each column of R corresponds to a class and each row of that column denotes the degree to which a class should be characterized (based on the training samples) by the corresponding subdomains. In other words, for a 3 class (denoted by C_1, C_2 and C_3) and 2 feature (denoted by F_1 and F_2) problem (i.e., $M = 3$ and $N = 2$) with 3 domains (denoted by a, b and c) corresponding to each feature space (i.e., with $\hat{N} = 3 \times \{3\} = 9$ subdomains), a relational matrix, R, can be written as follows.

$(F_1 \times F_2)$	C_1 C_2 C_3
(a, a) 1	r_{11} r_{12} r_{13}
(a, b) 2	r_{21} r_{22} r_{23}
(:)	:::
(c, c) 9	r_{91} r_{92} r_{93}

If a pattern belongs to the first subdomain, i.e., F_1 *is* a *and* F_2 *is* a then the entry r_{11} will denote the possibility value of the pattern to be in the class C_1. Similar is the case for all other entries of the relational matrix.

Determination of R: The relational matrix R is estimated from the training samples in the relational matrix estimator block. Let r_{hj} denotes the (h, j)th element of R, i.e., the element corresponding to the hth subdomain and jth pattern class. The value of r_{hj} is decided as

$$r_{hj} = \begin{cases} 0 & \text{if } h\text{th subdomain does not highlight } j\text{th pattern class;} \\ 1 & \text{if } h\text{th subdomain highlights only } j\text{th pattern class;} \\ (0.8)^{\frac{NS_h}{NG_h NC_j^h}} & \text{if } h\text{th subdomain highlights } j\text{th pattern class along with some other classes.} \end{cases} \quad (8)$$

Here NG_h is the number of training sample groups highlighted by the subdomain h; NC_j^h is the number of training samples from the jth class (C_j) in the hth subdomain and NS_h is the total number of training samples in the hth subdomain, i.e.,

$$NS_h = \sum_{j=1}^{M} NC_j^h.$$

If $NG_h = 0$ then $r_{hj} = 0$ for all $j = 1, 2, \cdots, M$. If $NG_h = 1$ and hth subdomain highlights the class C_j then $r_{hj} = 1$ and $r_{hk} = 0$ for $k \neq j$. Otherwise, if $NG_h > 1$, then the subdomain h is overlapping according to the training samples. The factor $NS_h/(NG_h NC_j^h)$ is used as a density factor for the jth pattern class in the hth (overlapping) subdomain.

So the block relational matrix estimator provides R, which is utilized in the fuzzy classifier block to find the output of the recognition system.

IV. Fuzzy Processor

This section consists of three parts, namely feature extractor, fuzzy classifier and decision maker. The feature extractor gives a characteristic vector $CV(X)$ as output corresponding to an input X. The $CV(X)$ along with the relational matrix is used in the fuzzy classifier to determine the degree of similarity of the input pattern X to the various pattern classes. The decision maker block gives a linguistic output along with its degree of certainty.

A. Feature Extractor

Here, the input patterns of the recognition system are in any of the four forms namely, *quantitative form*, *mixed form*, *set form*, and *linguistic form*. First of all, each feature value is considered separately to determine its membership values corresponding to various domains of the considered feature space. The way it has been done is furnished in the following subsection.

Quantitative Form: The information in this form are considered as in exact numerical terms, like "F_i *is* 500."

The membership functions corresponding to different domains of the individual feature spaces are decided in the preprocessing block (Section III). So for the information in exact numerical terms, the membership values to belong to different domains in the feature range in consideration are determined directly from the corresponding membership functions.

Mixed Form: The information are provided in this form as the mixture of linguistic hedges and quantitative terms such as "F_i is more or less 500."

As the linguistic hedges increase the impreciseness of the information, the membership values of a information in this form as a whole, for different domains should be lower than that of the membership values of the information with quantitative term alone. The amount of decrease is determined according to the linguistic hedges. As an example, for the information "F_i is about 500," the membership value corresponding to the gth domain in the ith feature space is assigned as

$$\mu_{ig}(F_i \text{ is about } 500) = \{\mu_{ig}(F_i \text{ is } 500)\}^{1.25} \quad (9)$$

where $\mu_{ig}(.)$ represents the membership value corresponding to the gth ($g = 1, 2, \cdots, n_i$) domain in the ith ($i = 1, 2, \cdots, N$) feature space.

The aforementioned modifications of the membership values will be reflected in the confidence factor (CF), i.e., in the final output of the recognition system.

Set Form: Like the mixed form, the information in *set form* are also a mixture of linguistic hedges and numerical terms. The basic difference lies with the nature of linguistic hedges used. The linguistic hedges used in this form are *less than*, *more than*, *between* etc., such that the data reflected is a set and at least one boundary of the data set becomes known. The example of the information in this form are "F_i is less than 500," "F_i is between 400 and 500" etc.

Initially, the membership values of the numerical terms corresponding to various domains are determined directly from the corresponding membership function that is of the form $\pi_{ig}(x, \alpha_{ig}, \beta_{l_{ig}}, \beta_{u_{ig}}, \Gamma_{l_{ig}}, \Gamma_{u_{ig}})$ (1) where α_{ig} is the central point of the gth domain in the ith feature space. For the statement F_i *is less than* v, the membership value corresponding to the gth domain of the ith feature space is decided as (10), shown at the bottom of the next page, where $\mu_{ig}(.)$ represents the membership value corresponding to the gth ($g = 1, 2, \cdots, n_i$) domain in the ith ($i = 1, 2, \cdots, N$) feature space.

For the linguistic hedges like *greater than* or *more than* where exactly one boundary of the reflected data set is known, the membership values are similarly decided.

There may be information with statements using the connectors *and*, *but* etc. (e.g., F_i is greater than 400 and/but less than 500) where the reflected data sets are both way bounded. In such cases, initially the two statements are considered separately and two membership values are determined. The resultant membership value is decided as the geometric mean of the two membership values, e.g.,

$$\mu_{ig}(F_i \text{ greater than } v_1 \text{ and less than } v_2) = [\mu_{ig}(F_i \text{ is greater than } v_1) \times \mu_{ig}(F_i \text{ is less than } v_2)]^{1/2}. \quad (11)$$

There may be statements like "F_i *is between* 400 and 500" which is equivalent to the statement "F_i is greater than 400 and less than 500" and proceed as in the previous case. Hence, to put it concisely, rules for the calculation of membership values can be found out if the provided information is in the *set form*.

Linguistic Form: The information provided in this form are completely in linguistic terms such as "F_i *is small*" or "F_i *is more or less high*."

To handle linguistic information, the system assumes only three primary linguistic variables, namely *small*, *medium* and *high* and the corresponding membership functions considered as $1 - S$, π and S functions respectively. Using the *a priori* knowledge, the values of the parameters of the membership functions are assigned.

As long as the membership functions are chosen properly, one recovers [21] the entirety of the classical logic for the designations of *true* and *false*. This implies that the system finds two truth values to indicate the interval of the truth values corresponding to a linguistic feature information. Here the system assumes for the two linguistic variables *small* and *high* that *true* $\equiv [0.5, 1]$, *false* $\equiv [0, 0.5]$ and extend this particular logic by adding

$$\begin{aligned} \text{very true} &\equiv [0.8, 1.0] \\ \text{more or less true} &\equiv [0.6, 0.8] \\ \text{neither true nor false} &\equiv [0.4, 0.6] \\ \text{more or less false} &\equiv [0.2, 0.4] \\ \text{very false} &\equiv [0.0, 0.2]. \end{aligned} \quad (12)$$

So corresponding to the previous type of interval-based truth value, one can find an interval of feature values that can be considered an equivalent of any linguistic information. That means, corresponding to a linguistic feature information, the system finds an interval of feature values. In other words, the system converts the linguistic information in *set form*. Then finds the membership value for various domains of the considered feature space depending on the converted information in *set form* from linguistic form.

The previous interval based truth value logic can not be directly used for the primary linguistic variable *medium*, whose membership function is a π function. Here the aforementioned interval based truth value reflects two different data sets and, accordingly, two different membership values for various domains are obtained. Finally, the maximum of these two membership values is retained as the membership value corresponding to each domain in the feature space.

It may happen that the information about a particular feature is fully unavailable or missing. In such cases, it is reasonable to assign some low (say, 0.2) membership value to all the feature domains in that particular feature space. It is done to keep the system's ability of handling the missing information, i.e., to decide the output based on the available partial (or incomplete) information. The logic behind assigning low membership values for missing information is to bring down the confidence of the system's output.

The aforementioned discussion shows the way, how the impreciseness/ uncertainty in the input feature information has been handled by providing/ modifying the membership values heuristically to a great extent. The logic behind the assignment of membership values is also intuitively appealing.

Characteristic Vector: The membership values corresponding to any input pattern X to be in the obtained subdomains are denoted by a vector, named as characteristic vector $CV(X)$. A typical pattern X consists of the individual feature information, i.e., $X = (F_1, F_2, \cdots, F_N)$. Initially, each individual feature information is considered separately to find the membership values to the domains of the individual feature space. The approaches to determine the membership values from the feature information are discussed previously.

Let us consider a typical, say hth ($h = 1, 2, \cdots, \hat{N}$), subdomain that consists of the following domains

$$(g_1^h, g_2^h, \cdots, g_i^h, \cdots, g_N^h) \quad (13)$$

where depending on h, g_i^h represents a particular domain in the ith feature space. Suppose $\mu_{g_i^h}(X)$ represents the membership of X to belong in the g_i^hth domain. So the hth element of $CV(X)$ i.e., the membership value corresponding to hth subdomain, is defined as the arithmetic mean of the membership values of individual feature domains, i.e.,

$$cv_h(X) = \begin{cases} \frac{1}{N}\sum_1^N \mu_{g_i^h}(X) & \text{if } \mu_{g_i^h}(X) > 0 \\ & \quad \text{for all } i = 1, 2, \cdots, N \\ 0 & \text{otherwise} \end{cases} \qquad h = 1, 2, \cdots, \hat{N}. \quad (14)$$

So the block feature extractor finds a characteristic vector with $\hat{N}$ elements (for $\hat{N}$ subdomains) corresponding to each input pattern X. This $CV(X)$ along with the relational matrix R are utilized in the fuzzy classifier to find the degree of similarity of the input X to the various pattern classes.

$$\mu_{ig}(F_i \text{ is less than } v) = \begin{cases} [\mu_{ig}(v)]^{1/2}, & \text{if } v \geq \alpha_{ig} \text{ and } \mu_{ig}(v) > 0 \\ [\mu_{ig}(v)]^2, & \text{if } v \leq \alpha_{ig} \text{ and } \mu_{ig}(v) > 0 \\ 0.2, & \text{if } v > \alpha_{ig} \text{ and } \mu_{ig}(v) = 0 \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

B. Fuzzy Classifier

In 1977, Zadeh [8] suggested the compositional rule of inference for the fuzzy conditional implication. Although, other authors [9]–[11] have suggested different methods, we have restricted to Zadeh's compositional rule of inference for developing the system. It is defined here before describing the classifier.

Definition 1: Let $\mathcal{A}$ denote a fuzzy set in $\mathcal{X}$ and $\mathcal{R}$ denote a fuzzy relation in $\mathcal{X}\times\mathcal{Y}$. Then the compositional rule of inference asserts the solution of the relational assignment equations [19]

$$\mathcal{R}(x) = \mathcal{A} \text{ and } \mathcal{R}(x, y) = \mathcal{B}$$

is given by

$$\mathcal{C} = \mathcal{R}(y) = \mathcal{A} \bigcirc \mathcal{B} = \max_x \min\{\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x, y)\} \quad (15)$$

where $\mathcal{A} \bigcirc \mathcal{B}$ is the $\max - \min$ composition of $\mathcal{A}$ and $\mathcal{B}$.

Note: The $\min$ operator finds the minimum of any two elements, i.e., it provides the connective information. When the minimum of the two elements is kept same but the value of the other element is increased, the effect is not reflected by the $\min$ operator. On the other hand, the arithmetic mean (AM) operator finds the middle most value of any two elements, i.e., it gives collective information. Any change in any of the elements is reflected by the *AM* operator.

The classifier incorporates the previous $\max - \min$ compositional rule of inference in a modified way. The *min* operator of $\max - \min$ operation in (15) is replaced by *AM* operator. That is, the classifier incorporates the *max-AM* compositional rule of inference. So the class similarity vector $S(X)$ is determined as (16) (shown at the bottom of the page), where $cv_h(X)$ is the hth element of $CV(X)$; r_{hj} is the (h, j)th entry of R and $\hat{N}$ is the number of subdomains. Hence the block fuzzy classifier finds a class similarity vector $S(X)$ corresponding to an unknown input X.

Example 1: Suppose there are obtained 4 subdomains (denoted by SD_1, SD_2, SD_3 and SD_4) for a 3 class (denoted by C_1, C_2 and C_3) problem. Let the characteristic vector be $CV(X) = [0.7, 0.3, 0.0, 0.0]$ for an input pattern X and the relational matrix R is estimated as follows.

F	C_1 C_2 C_3
SD_1	0.0 1.0 0.0
SD_2	1.0 0.0 0.0
SD_3	0.6 0.0 0.9
SD_4	0.0 0.0 1.0

The similarity vector $S(X)$ for the pattern X will be

$$S(X) = CV(X) \bigcirc R = [0.65\ 0.85\ 0.0].$$

Here $S(X)$ indicates that the unknown pattern X is inclined to the class C_2.

C. Decision Maker

The similarity vector $S(X)$ is analyzed in the decision-maker block. The system always tries to provide multiple output choices for classes with their preferences. That is, the outputs will be one of the following types:

1) *Single Choice*: If the entry in $S(X)$ corresponding to only one class, say C_j, is positive then the class C_j is considered as the output with *single choice.*
2) *Combined Choice*: If the entries in $S(X)$ corresponding to more than one class are positive and are nearly same (difference ≤ 0.05) then the said classes are considered as output with *combined choice.*
3) *First-Second Choice*: If the entries in $S(X)$ corresponding to at least two classes are positive and the said entries do not satisfy the criteria for *combined choice* then *first-second choice* is considered. The highest two entries in $S(X)$ are taken as the *first* and *second choices* respectively.
4) *Null choice*: If all the entries in $S(X)$ are zero then the system refuses to assign the unknown sample to any class, i.e., *null choice* is given.

It is to be mentioned here that the *single choices* estimate the nonoverlapping regions in the feature space whereas the *combined* and *first-second choices* estimate the overlapping regions. *Null choices* estimate the portions uncovered by the training samples (with extended portions) and also the portions not represented by any class.

In order to give the final output decision in linguistic form regarding the class or classes to which the unknown input pattern X may belong, a measurement of confidence factor (CF) is defined as

$$\mathrm{CF} = \frac{1}{2}\left[s_{mod}(X) + \frac{1}{M-1}\sum_{j=1}^{M}\{s_{\mathrm{mod}}(X) - s_j(X)\}\right]$$

$$0 \leq \mathrm{CF} \leq 1 \quad (17)$$

where $s_{mod}(X)$ is the highest entry in $S(X)$; $s_j(X)$ denotes the jth entry in $S(X)$ and M denotes the number of classes.

For the case of *single choice*, the linguistic variable *surely* is attached to the final output decision. Otherwise the linguistic variable *likely* is attached to the output. Based on the CF values, the linguistic hedges *very*, *more* or *less*, *not* etc. are

$$S(X) = CV(X) \bigcirc \mathcal{R} = \begin{cases} \max_{h=1,2,\cdots,\hat{N}}\left\{\frac{1}{2}(cv_h(X) + r_{hj})\right\}, & \text{if } cv_h(X) > 0 \text{ and } r_{hj} > 0 \\ 0, & \text{otherwise} \\ & j = 1, 2, \cdots, M. \end{cases} \quad (16)$$

assigned with the linguistic output decision as follows:

1) very true : if $0.8 \leq \text{CF} \leq 1.0$
2) true (only) : if $0.6 \leq \text{CF} < 0.8$
3) more or less true : if $0.4 \leq \text{CF} < 0.6$
4) not false : if $0.0 < \text{CF} < 0.4$.

In the case of *null choice*, the system gives the linguistic output decision as *unable to recognize*. The CF values are always attached along with the linguistic output decision.

Some typical output forms are:

1) This is *very surely* to be C_1 (CF= 0.89).
2) This is *likely* to be C_1 (CF= 0.72) but *not unlikely* to be C_2 (CF= 0.32).
3) This is *more or less likely* to be C_1 (CF= 0.48) but *not unlikely* to be C_2 (CF= 0.25).
4) This is *not unlikely* to be C_1 (CF= 0.28).
5) This is *more or less likely* to be either C_1 or C_2 (CF= 0.52).

V. Implementation and Results

To verify the effectiveness of the proposed recognition system, different possible pattern sets were first of all generated and the previous algorithm was implemented on them. The recognition scores are found to be quite satisfactory in all the cases. Figs. 6(a)–(d) show typical pattern sets in two dimensional feature space. In Fig. 6(a), there are six pattern classes (denoted by A, B, C, D, E, and F respectively) with 120, 120, 90, 90, 180 and 120 samples respectively. In Fig. 6(b), there are three pattern classes (denoted by A, B, and C respectively) with 300, 100 and 100 samples respectively. In Fig. 6(c), there are two pattern classes (denoted by A and B) with 100 samples in each class. In Fig. 6(d), there are two pattern classes (denoted by A and B) with 100 and 150 samples respectively.

To implement the proposed algorithm, five different sets of 10% training samples were chosen randomly from each of the previous four sets of pattern classes. The recognition scores for the considered four cases are shown in Tables I-A through I-D, respectively. The scores shown are obtained by averaging those corresponding to five different training sets. The membership functions of various domains along the feature axes are considered as the π functions ((1)). Assuming the linear triangular membership functions ((3)), more or less same results are obtained. Note that the recognition scores are grouped into five categories, namely *single correct choice*, *first correct choice*, *combined correct choice*, *second correct choice* and *fully wrong choice*. The *single correct choice* set includes those samples for which the system's *single choice* corresponds to the actual class. The *first correct choice* set includes those samples for which the system provides *first-second choice* with first choice as the actual class. The *combined correct choice* set includes those samples for which the classifier provides *combined choice* and one of the choices corresponds to the actual class. The *second correct choice* set includes those, for which the system considers *first-second choice* with second choice as the actual class. Samples not falling under the aforementioned categories are termed as *misclassification* or *fully wrong choice*. It is to be noticed that the *first-second choices* provide the states *first correct* and *second correct choices*. Hence the four output forms of the recognition system are categorized in the aforementioned five states.

Observe that the pattern classes in Fig. 6(a) are of regular (elliptical) shape and there exists overlapping between two or more classes. On the other hand, the pattern classes in Fig. 6(b)–(d) are irregular shape and they are mutually nonoverlapping. In these three cases, most (95.4% to 99.6%) of the samples are seen to be recognized by *single choices* and the remaining samples are recognized either by *first* or *combined choices*. There is no samples falling under the sets *second* and *fully wrong choices*. In case of Fig. 6(a), there are some samples that are found under the sets *second correct* and *fully wrong choices*. It is to be noted that when the overlapping is between two classes then the samples are recognized either by *single choice* or *first choice* or *combined choice* or *second choice* and so there will not be any sample falling under the set *fully wrong choice*. In case the overlapping exists between more than two classes, some samples will obviously fall under *fully wrong choice*. In such cases, it may be possible to avoid this situation by providing a higher choice namely, *third choice*.

To examine the practical applicability, the algorithm was then implemented on a set of *Indian Telugu Vowel Sounds* in a consonant-vowel-consinant context uttered by three speakers in the age group 30 to 35 years. Fig. 7 shows the typical feature space in $F_1 \times F_2$ plane of the six vowels (δ, a, i, u, e, o) containing 871 samples. F_1 and F_2 denote the first and second formant frequencies that were obtained through spectrum analysis of the speech data. The boundaries of the classes are seen to be ill-defined (fuzzy). The details of the feature extraction procedure is available in [12].

The test set consists of the aforementioned 871 data and 102 imprecise (incomplete) data. These imprecise data on F_1 and F_2 were coded to various linguistic forms viz., (700, *between* 1800 *to* 2200), (*about* 600, *more or less high*), (*small*, –) etc. by the trained personnel. It is to be mentioned here that these imprecise samples were ignored in earlier works [22]–[26] that were incapable of handling them. The recognition score of the vowel data is shown in Table II where the classifier is trained with a set of 10% samples drawn randomly from 871 data. The membership functions of various domains along the feature axes are considered here as the π functions. Assuming the linear triangular membership functions, similar results are obtained.

A list of some typical output is given for illustration.

1) (300, 900): This is *surely* to be u (CF= 0.78).
2) (700, 1300): This is *more or less likely* to be a (CF= 0.55) but *not unlikely* to be δ (CF= 0.32).
3) (850, 1300): This is *more or less sure* to be a (CF= 0.25).
4) (250, 1550): *unable to recognize* this sample.
5) (*between* 500 and 650, 1600): This is *likely* to be i (CF= 0.63) but *not unlikely* to be e (CF= 0.32).

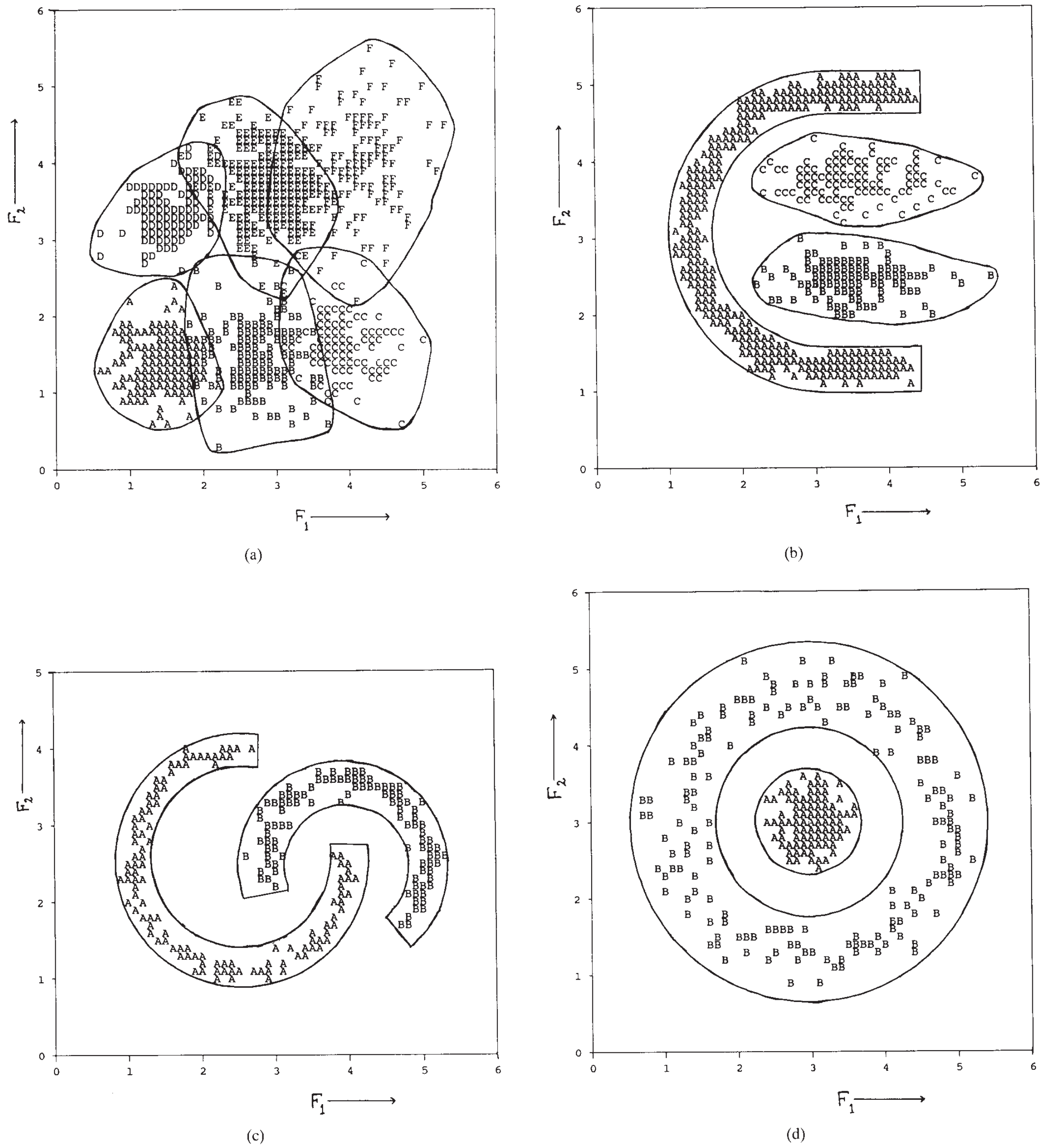


Fig. 6. (a)–(d) Show four sets of pattern classes.

6) (*about* 350, –): This is *likely* to be either i or u (CF= 0.28).

These natural outputs confirm the vowel diagram in Fig. 7. Note that for the input (250, 1550), the system is unable to recognize the vowel, as this information is having very much insignificant similarity with the vowel classes. This has been regarded as *misclassification* while computing the recognition score. Further, for the information (*about* 350, –) (here "–" indicates that there is no information for F_2 feature), the system finds some similarity with the vowel classes i and u, on the basis of the F_1 feature information alone.

VI. Conclusion

A recognition system having the flexibility of accepting input in *quantitative form*, *linguistic form*, *set form* and *mixed form*, and in providing output decision in natural (linguistic)

TABLE I-A
RECOGNITION SCORE FOR THE PATTERN CLASSES IN FIG. 6(a)

	%Recognition Score						
	Actual Classes						Overall
Various Group of Choices	A	B	C	D	E	F	Score
Single Correct Choice	75.00	50.38	77.78	66.67	72.79	67.50	68.67
First Correct Choice	19.17	24.17	16.66	17.77	22.77	16.67	19.80
Combined Correct Choice	4.17	7.5	1.11	6.67	2.22	3.33	4.03
Second Correct Choice	1.66	16.67	3.33	8.89	1.67	11.67	6.94
Fully Wrong Choice	0.00	0.83	1.11	0.00	0.56	0.83	0.56

TABLE I-B
RECOGNITION SCORE FOR THE PATTERN CLASSES IN FIG. 6(b).

	%Recognition Score			
	Actual Classes			Overall
Various Group of Choices	A	B	C	Score
Single Correct Choice	95.33	94.00	97.00	95.40
First Correct Choice	3.00	3.00	2.00	2.80
Combined Correct Choice	1.66	3.00	1.00	2.80
Second Correct Choice	0.00	0.00	0.00	0.00
Fully Wrong Choice	0.00	0.00	0.00	0.00

TABLE I-C
RECOGNITION SCORE FOR THE PATTERN CLASSES IN FIG. 6(c)

	%Recognition Score		
	Actual Classes		Overall
Various Group of Choices	A	B	Score
Single Correct Choice	98.00	100.00	99.00
First Correct Choice	1.00	0.00	0.50
Combined Correct Choice	1.00	0.00	0.50
Second Correct Choice	0.00	0.00	0.00
Fully Wrong Choice	0.00	0.00	0.00

TABLE I-D
RECOGNITION SCORE FOR THE PATTERN CLASSES IN FIG. 6(d)

	%Recognition Score		
	Actual Classes		Overall
Various Group of Choices	A	B	Score
Single Correct Choice	98.00	100.00	99.60
First Correct Choice	1.00	0.00	0.40
Combined Correct Choice	0.00	0.00	0.00
Second Correct Choice	0.00	0.00	0.00
Fully Wrong Choice	0.00	0.00	0.00

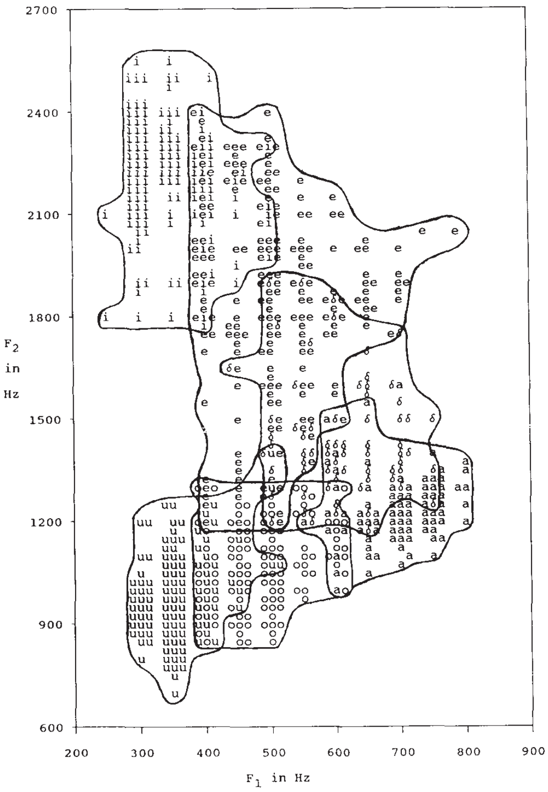


Fig. 7. Vowel classes in the $F_1 \times F_2$ plane.

form along with its degree of certainty has been formulated. In order to show the effectiveness of the proposed system, different artificial pattern sets were considered. Problem of recognizing vowel sound in consonant-vowel-consonant context has also been considered to demonstrate its practical applicability to real life data. The results shown in Tables I and II using the π membership functions. The results obtained using the linear triangular membership functions ((3)), are more or less same.

The recognition system always tries to provide multiple class choices and classify a sample either as *single* or *first-second* or *combined* or *null choices*. The recognition system has the capability of reflecting the overlapping and nonoverlapping regions. The *single choices* estimate the nonoverlapping regions. The overlapping regions are estimated by *first-second* and *combined choices*. The *null choices* reflect the portions outside the pattern classes and/or the portions uncovered by the training samples or by the obtained feature domains (with the extended portions).

The decisions of the existing conventional classifiers can be categorized into *two hard states* as *correct* or *wrong*. Each of these classifiers is designed to apply to some particular situations. If they are applied in the situations, different from the aimed ones, then satisfactory results, in general, are not obtained. Moreover the conventional classifiers can not provide multiple class choices and so they can not estimate the overlapping or nonoverlapping portions of the pattern classes in the feature space.

In contrast to the conventional classifiers, the proposed recognition system can be applied to all possible situations. It

TABLE II
RECOGNITION SCORE FOR THE VOWEL CLASSES IN FIG. 7

	%Recognition Score						
	Actual Classes						
Various Group of Choices	δ	a	i	u	e	o	Overall Score
Single Correct Choice	40.05	59.47	68.72	67.92	52.19	55.37	58.92
First Correct Choice	11.34	22.55	24.88	26.12	20.27	22.41	21.76
Combined Correct Choice	20.83	11.24	2.91	1.99	13.04	8.33	8.63
Second Correct Choice	19.44	6.74	3.49	3.97	12.56	13.89	9.56
Fully Wrong Choice	8.34	0.00	0.00	0.00	1.94	0.00	1.13

is to be observed that the proposed algorithm does not assume any distribution of the pattern classes. Only assumption it made is that the training samples more or less should represent the classes. The effectiveness of most of the existing classifiers depends on the distribution of the pattern classes. Bayes classifier is the most well known and established classifier, and we have tried to apply this on the artificially generated pattern sets for the comparison purpose. If the classes are of regular shaped and if their distributions can be obtained nicely, the performance of the Bayes classifier is more or less same with our system (considering *single*, *first* and *combined correct choices*). For example, the classes in Fig. 6(a) are of regular (elliptical) shaped and the recognition score of the Bayes classifier was found to be 90.64 whereas, the recognition score for our classifier with *single*, *first* and *combined correct choices* is 92.50. Again analyzing the results, it has been found that our output decisions showing multiple choices are more natural and justified.

When the pattern classes are not of regular shaped, it is extremely difficult to find their distributions. In such cases, multivariate normal distributions are assumed. But the Bayes classifier with multivariate normal distributions gives poor result, or in other word it may not always be applied on such pattern classes. For example, we could not apply the Bayes classifier on the pattern classes in Fig. 6(b)–(d), whereas it is not difficult to apply the proposed algorithm on these data sets, and the recognition scores (Tables I-B and I-D) were found to be very satisfactory.

It is observed from the vowel recognition problem that the confusion in recognizing a sample considering the *single* and *first choices* lies, in general, only with the neighboring classes constituting a vowel triangle. The similar findings were also obtained with the previous investigations [22]–[26], considering deterministic input/output. The overall recognition score is quite satisfactory considering the fact that it accepts approximate feature information and the information relates only F_1 and F_2. Feature F_3, which were incorporated in [22]–[26], has not been considered here.

The linguistic output decisions of the recognition system can be categorized in five states, namely, *single correct*, *first correct*, *combined correct*, *second correct* and *fully wrong (null)* choices. For a sample with *first-second choice*, if the first choice corresponds to the actual class then it is included in the *first correct* choice set, and if the second choice corresponds the correct class then it is included in the *second correct* choice set. Hence the considered four output forms of the system are categorized in the aforementioned five states. This is explained in Fig. 8. Because of the flexibility, the system has a provision of improving its efficiency significantly by incorporating *combined* and *second choices* under the control of a supervisory scheme.

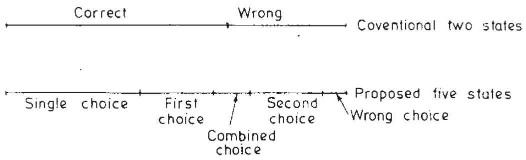


Fig. 8. Conventional two state versus proposed five state output.

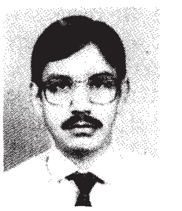
ACKNOWLEDGMENT

The authors gratefully acknowledge Prof. D. Dutta Majumder for his interest in this work and Mr. S. Chakraborty for drawing the diagrams. One of the authors (S.K. Pal) is grateful to the National Research Council, U.S., to provide him an NRC-NASA Senior award to work at the Johnson Space Center, Houston, TX. The authors would like to thank the anonymous referees for their constructive comments that greatly helped in revising the manuscript.

REFERENCES

[1] W. Pedrycz, "Fuzzy sets in pattern recognition: Methodology and methods," *Pattern Recog.*, vol. 23, pp. 121–146, 1990.
[2] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[3] K. S. Fu, *Syntactic Pattern Recognition and Applications*. London: Academic, 1982.
[4] L. A. Zadeh, "The role of fuzzy logic in the management of uncertainty in expert system," *Fuzzy Sets and Systems*, vol. 11, pp. 199–223, 1983.
[5] R. Martin-Clouaire and H. Prade, "On the problems of representation and propagation of uncertainty in expert systems," *Int. J. Man-Machine Studies*, vol. 22, pp. 251–264, 1985.
[6] L. A. Zadeh, "Fuzzy sets," *Inform. and Contr.*, vol. 8, pp. 338–353, 1965.
[7] A. Kaufmann, *Introduction to the Theory of Fuzzy Subsets—Fundamental Elements*. New York: Academic, 1975.
[8] L. A. Zadeh, "Fuzzy logic and approximate reasoning," *Synthese*, vol. 30, pp. 407–428, 1977.
[9] J. F. Baldwin, "a new approach to approximate reasoning using a fuzzy logic," *Fuzzy Sets and Systems*, vol. 2, pp. 309–325, 1979.
[10] R. R. Yager, "Approximate reasoning and possibility model in classification," *Int. J. Comp. Inf. Sci.*, vol. 10, pp. 141–175, 1981.
[11] S. K. Pal and D. P. Mandal, "Fuzzy logic and approximate reasoning: An overview," *J. Inst. Elec. Telecom. Engrs*, vol. 37, pp. 548–560, 1991.
[12] S. K. Pal and D. D. Majumder, *Fuzzy Mathematical Approach to Pattern Recognition*. New York: Wiley /Halsted, 1986.

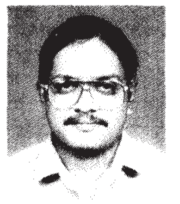
[13] A. Kandel, *Fuzzy Mathematical Techniques with Applications*. Reading, MA: Addison Wesley, 1986.
[14] R. R. Yager, "Multiple objective decision using fuzzy subsets," *Int. J. Man-Machine Studies*, vol. 9, pp. 375–382, 1977.
[15] M. M. Gupta, A. Kandel, W. Bandler, and J. B. Kiszka, *Approximate Reasoning in Expert Systems*. New York: North Holland, 1985.
[16] E. Sanchez, "Medical diagnosis and composite fuzzy relations," *Advances in Fuzzy Set Theory and Applications*, M. M. Gupta, R. K. Ragade and R. R. Yager, Eds. Amsterdam: North Holland, pp. 437–444, 1979.
[17] R. N. Lea and Y. Jani, "Fuzzy logic in autonomous orbital operation," *Int. J. Approximate Reasoning*, vol. 6, pp. 151–184, 1992.
[18] D. P. Mandal, C. A. Murthy, and S. K. Pal, "Determining the shape of a pattern class from sampled points in $\mathbb{R}^2$," *Int. J. Gen. Syst.*, vol. 20, pp. 307–339, 1992.
[19] L. A. Zadeh, K. S. Fu, K. Tanaka, and M. Shimura, *Fuzzy Sets and Their Application to Cognitive and Decision Process*. London: Academic, 1975.
[20] U. Grenander, *Abstract Inference*. New York: Wiley, 1981.
[21] D. G. Schwartz, "The case for an interval based representation of linguistic truth," *Fuzzy Sets Syst.*, vol. 17, pp. 153–165, 1985.
[22] A. K. Datta, N. R. Gangully, and S. Ray, "Maximum likelihood methods in vowel recognition: A comparative study," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 683–689, 1982.
[23] S. K. Pal and D. D. Majumder, "Fuzzy sets and decision making approaches in vowel and speaker recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 625–629, 1977.
[24] S. K. Pal, "Optimum guard zone for self supervised learning," *IEEE Proc. Part E*, vol. 129, pp. 9–14, 1982.
[25] S. K. Pal, A. Pathak, and C. Basu, "Dynamic guard zone for self supervised learning," *Patt. Recog. Lett.*, vol. 7, pp. 135–144, 1988.
[26] A. Pal(pathak) and S. K. Pal, "Effect of wrong samples on the convergence of learning Process-II," *Inform. Sci.*, vol. 60, pp. 77–105, 1992.



Deba Prasad Mandal received the B.Sc. (honors) degree in 1984 from Kalyani University, West Bengal, India, and the Master of Computer Applications degree in 1988 from Jawaharlal Nehru University, New Delhi, India. Presently, he is working toward the Ph.D. degree at the Electronics and Communications Sciences Unit at the Indian Statistical Institute, Calcutta, India.

He is a Senior Research Fellow in the Electronics and Communications Sciences Unit at the Indian Statistical Institute. His research interest mainly include pattern recognition, image processing, remote sensing, and fuzzy sets and systems.

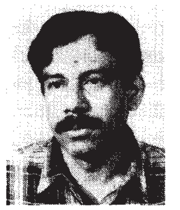
Mr. Mandal is a member of the Indian Society for Fuzzy Mathematics and Information Processing (ISFUMIP) and the Indian Unit for Pattern Recognition and Artificial Intelligence (IUPRAI). He is a life member of the Indian Science Congress Association. He received the Young Scientist Award in computer sciences for 1992 from Indian Science Congress Association.



C. A. Murthy was born in 1958 in Ongole, India. He received the B.Stat. (honors) degree in 1979, the M.Stat. degree in 1980, and the Ph.D. degree in 1989, all from the Indian Statistical Institute, Calcutta, India.

He is working as a Lecturer in the Electronics and Communication Sciences Unit at the Indian Statistical Institute. His research interests are in pattern recognition, image processing, computer vision, fuzzy sets, and remote sensing.

Dr. Murthy is a member of the Indian Society for Fuzzy Mathematics and Information Processing (ISFUMIP) and the Indian Unit for Pattern Recognition and Artificial Intelligence (IUPRAI).



Sankar K. Pal (M'81–SM'84) received the B.Sc. (honors) degree in physics and the B.Tech., M.Tech., and Ph.D. degrees in radiophysics and electronics in 1969, 1972, 1974, and 1979, respectively, from the University of Calcutta, Calcutta, India. He received the Ph.D. in electrical engineering along with the Diploma of Imperial College in 1982 from Imperial College, University of London, London, England.

He is a Professor in the Electronics and Communications Sciences Unit at the Indian Statistical Institute, Calcutta, India. He was a recipient of the Commonwealth Scholarship in 1979 and MRC (U.K.) Postdoctoral Award in 1981 to work at Imperial College. In 1986 he was awarded the Fulbright Postdoctoral Visiting Fellowship to work at the University of California, Berkeley, and the University of Maryland, College Park. In 1989 he received an NRC-NASA Senior Research Award to work at the NASA Johnson Space Center, Houston, TX. He received the 1990 Shanti Swarup Bhatnagar Prize in Engineering Sciences for his contributions in pattern recognition. Presently, he is working as a Guest Investigator in the Software Technology Branch, NASA Johnson Space Center, Houston, TX. He served as a Professor-in-Charge of the Physical and Earth Sciences Division, Indian Statistical Institute, Calcutta, India, from 1988 to 1990. He was also a Guest Lecturer from 1983 to 1986 in Computer Science at Calcutta University. His research interests are mainly in pattern recognition, image processing, artificial intelligence, neural nets, and fuzzy sets and Systems. He is a co-author of the book *Fuzzy Mathematical Approach to Pattern Recognition* (Wiley/Halsted, 1986), and he is a co-editor of the book *Fuzzy Models for Pattern Recognition* (IEEE Press, 1992). He has written more than 150 research papers, ten of which are in books and more than 90 in international journals. He has lectured on his research work at various universities and laboratories in the United States and Japan. He is listed in *Reference Asia, Asia's Who's Who of Men and Women of Achievements*.

Dr. Pal is an Associate Editor of *International Journal of Approximate Reasoning* (North Holland), a Reviewer of the Mathematical Reviews (American Mathematical Society), a Fellow of the IETE, a Life Member of the Indian Statistical Institute and Treasurer of the Indian Society for Fuzzy Mathematics and Information Processing (ISFUMIP). He is also a Permanent Member of the INDO-US Forum on Cooperative Research and Technology Transfer (IFCRTT) and Organizing/Program Committee Member of various International Conferences and Meetings.