UNIT 1 FUZZY SETS – AN INTRODUCTION

Structure

1.1 INTRODUCTION

Fuzzy sets are a further development of the mathematical concept of a set. Sets were first studied formally by the German mathematician Georg Cantor (1845-1918). Even though the idea of fuzzy sets was introduced in its modern form by Lotfi A. Zadeh in 1965, the idea of multi-valued logic in order to deal with vagueness has been around from the beginning of the century. Fuzzy set theory generalizes classical set theory to allow partial membership in the sense that the membership degree of an object to a set is not restricted to first the integers 0 and 1, but may take on any value in the interval $[0,1]$.

In this unit the basic concepts and the essence of fuzzy sets are presented in Section 1.2. The fuzzy set operations and the fuzzy relation are also discussed. At the end, some applications of the fuzzy sets for creating fuzzy logic systems are presented. To get an idea about the essence of a fuzzy set, a simple example is presented below. The best way to introduce fuzzy sets is to start with a limitation of classical set. A set in classical set theory always has a sharp boundary because membership in a set is a black-and-white concept, i.e., an object either completely belongs to the set or does not belong to the set at all.

For example, suppose we wish to represent the set of high-income families within the framework of classical set theory. This would require us to choose a particular value as a threshold (e.g., yearly income) such that all families with a yearly income greater than the threshold (say, Rs. 2,500,00) are considered to be a member of the set of high-income families, and all families with a yearly income lower than the threshold are not members of the set. However, the fact that a family with a yearly income of Rs. 2,499,99 is not considered high income whereas a family with an extra Rs. 1 in the yearly income is considered in the high income group is counterintuitive, to say the least. Even though unnatural and sometimes disturbing, such artificially created sharp boundaries have often been used in defining classical sets, primarily due to the lack of a better methodology.

Even though some sets do have sharp boundaries (e.g., the set of married people), many others do not have sharp boundaries (e.g., the set of happily married couples, the set of good graduates, the set of tall persons, etc.). Fuzzy set theory directly addresses this limitation by allowing membership in a set to be a matter of degree. The degree of membership in a set is expressed by a number between 0 and 1;0 means entirely not in the set, 1 means completely in the set, and a number in between means partially in the set. This way, a smooth and gradual transition from the region outside the set to those in the set can be described. In Section 1.3 and 1.4, Support and α -cut and hedges are

discussed respectively. We shall discuss the operations on fuzzy set in Section 1.5 and fuzzy relation in Section 1.6.

Objectives

After studying this unit, you should be able to

- know the essence of the concept of fuzzy sets
- define the fuzzy set
- differentiate the classical set and the fuzzy set
- apply the theory of fuzzy set to real-life complex problems
- design membership functions for fuzzy sets
- apply operations on fuzzy sets.
- define fuzzy relations

1.2 CLASSICAL SET VS. FUZZY SET

Let us start with the formal definition of classical sets.

Definition 1 (Classical Set): Any collection of objects which can be treated as an entity is known as classical set. Cantor described a set by its members, such that an item from a given universe is either a member or not. The terms *set*, *collection* and *class* are synonyms, just as the terms *item*, *element* and *member are*.
There are three basic methods by which sets can be defined within a given universal set X:

(a) **List method** – This method defines a set by listing all its members. Since, it is not possible to list all the elements of an infinite set; this method is generally confined to define only finite sets. Set A, whose members are $a_1, a_2, \cdots, a_n$, is usually written as

$$A = \{a_1, a_2, \cdots a_n\}$$

(b) **Rule method –** In this method, a set is defined by a property satisfied by its members**.** A common notation expressing this method is

$$A = \{x | P(x)\} \text{ or } A = \{x : P(x)\}$$

Where the symbol 'l' or ':' denote the phrase "such that", and $P(x)$ designates a preposition of the form "x has the property P". That is, A is defined in this notation as the set of all elements of χ for which preposition P(x) is true. It is required that the property P be such that for any given $x \in \chi$, the preposition P(x) is either true or false.

(c) **Characteristic function method** – This method defines a set by using a function, called a characteristics function that declares which elements of A are members of the set and which are not. A set A is defined by its characteristics function, χ_A, as follows

$$\chi_A(x) = \begin{cases} 1 & for\ x \in A \\ 0 & for\ x \notin A \end{cases}$$

That is, the characteristics function maps elements of A to elements of the set $\{0,1\}$, which is formally expressed by

$$\chi_A : A \to \{0,1\}$$

For each $x \in A$, when $\chi_A(x) = 1$, x is declared to be a member of A; when $\chi_A(x) = 0, x$ is declared as a nonmember of A.

Example 1 (classical sets): The following are well defined lists or collections of objects, and therefore entitle to be called sets:

(a) *The set of non-negative integers less than 4. This is a finite set with four members: 0, 1, 2 and 3.*

(b) *The set of live dinosaurs in the basement of the British Museum. This set has no members and is called an empty set or a null set.*

(c) *The set of measurements greater than 10 volts. Even though this set is infinite, it is possible to determine whether a given measurement is a member or not.*

Now, try an exercise.

E1) List the elements of the following sets described by rule method:

(i) $A = \{n \in N : n \text{ is an odd prime less than } 150\}$

(ii) $B = \{x \in R : x^2 - 8x + 15 = 0\}$

(iii) $C = \{\max\{a,b\} : a, b \text{ are twin primes } \leq 150\}$

[Note: two consecutive primes are said to be twin if their difference is 2, e.g., <3, 5>, <5, 7>, <11, 13> are all twin-primes.**]**

Now let us define the fuzzy set formally.

Fuzzy Sets: According to Zadeh, many sets have more than an either-or criterion for membership. Take for example the set of *young people*. A one year old baby will clearly be a member of the set, and a 100 years old person will not be a member of this set, but what about people at the age of 20, 30, or 40 years? Another example is a weather report regarding high temperatures, strong wind, or nice days. In other cases a criterion appears non-fuzzy, but is perceived as fuzzy: a speed limit of 60 kilometers per hour, a check-out time at 12 noon in a hotel, a 50 years old man. Zadeh proposed a *grade of membership*, such that the transition from membership for all its members thus describes a fuzzy set. An item's grade of membership is normally a real number between 0 and 1, often denoted by the Greek letter μ. The higher the number, the higher the membership. Zadeh regards Cantor's set a special case where elements have full membership, i.e., $\mu = 1$. He nevertheless called Cantor's sets *non-fuzzy*; today the term *crisp* set is used, which avoids that little dilemma. Notice that Zadeh does not give a formal basis for how to determine the grade of membership. The membership value for a 50 year old in the set *young* depends on one's own view. The grade of membership is a precise, but subjective measure that depends on the context.

Following the above discussions, a fuzzy set is thus defined using a function that maps objects in a domain of concern to their membership value in the set. Such a function is called the *membership function* and is usually denoted by the Greek symbol μ for ease of recognition and consistency. For example, a more realistic representation of the high-income family can now be expressed by the membership function shown in figure 1. The membership function of a fuzzy set A is denoted as μ_A, and the membership value of x in A is denoted by $\mu_A(x)$. The domain of the membership function, which is the domain of concern from which the elements of the set are drawn, is called the *universe of discourse* and is denoted by U. For instance, the universe of discourse of the fuzzy set high-income can be the positive real line $[0, \alpha)$.

In addition to membership functions, a fuzzy set is also associated with a linguistically meaningful term. For instance, the fuzzy set in figure 1 is associated with the linguistic term "high". Associating a fuzzy set to a linguistic term offers two important benefits. First, the association makes it easier for human experts to express their knowledge using the linguistic terms. Second, the knowledge expressed using linguistic term is easily comprehensible. This benefit often results in significant savings in the cost of designing, modifying, and maintaining a fuzzy logic system. An important concept in fuzzy set that enables these two benefits is that of the linguistic variable which is defined in the following paragraphs.

In summary, a fuzzy set has a dual representation: a qualitative description using a linguistic term and a quantitative description through a membership function, which maps elements in a universe of discourse (i.e., a domain of interest) to their membership degree in the set. The above two are represented by the concepts of "granulation" and "graduation", respectively.

Figure 1 – Membership function of high annual income

Figure 2 – Fuzzy set to represent "*comfortable house for a 4-person family*"

***Example 2 (fuzzy set)*:** Classifying houses problem – A realtor wants to classify the house he offers to his clients. One indicator of comfort of these houses is the number of bedrooms in them. Let the available types of houses of represented by the following set.

$$U = \{1,2,3,4,5,6,7,8,9,10\}$$

The houses in this set are described by μ number of bedrooms in a house. The realtor wants to describe a "*comfortable house for a 4-person family,*" using a fuzzy set.

***Solution*:** The fuzzy set to describe *"comfortable house for a 4-person family,"* may be defined in the following manner: *comfortable house for a 4-person family* = $0.2/1+0.5/2+0.8/3+ {}^{1}\!/_{4} +0.7/5+0.3/6$, which means that the level of comfort for a 1-bedroom house is 0.2, a 2-bedroom house is 0.5 and so on.

The equivalent graphical representation of it is shown in figure 2.

Try the following exercises.

E2) Describe the concept of a fuzzy set in your own words.

E3) Find at least five examples of fuzzy sets in newspaper articles or magazines.

Now let us discuss the various component related to fuzzy sets.

(i) **Linguistic Variables:** A linguistic variable is a variable whose values are an expression involving fuzzy sets. In other words, a linguistic variable is like a composition of a symbolic variable (a variable whose value is a symbol) and a numeric variable (a variable whose value is a number). An example of a symbolic variable is "Shape = Rectangular" where "Shape" is a variable indicating the shape of an object. An example of numeric variable "Height = 5'', where "Height" is a variable representing the height of a person. Numeric variables are frequently used in science, engineering, mathematics, medicine and many other disciplines, while symbolic variables play an important role in artificial intelligence and decision sciences.

A linguistic variable enable its value to be described both qualitatively by a linguistic term (i.e., a symbol serving as the name of a fuzzy set) and quantitatively by a corresponding membership function (which expresses the meaning of the fuzzy set). The linguistic term is used to express concepts and knowledge in human communication, whereas membership function is useful for processing numeric input data.

Using the notion of the linguistic variable to combine these kinds of variables into a uniform framework is, in fact, one of the main reasons the fuzzy logic has been successful in offering intelligent approaches in engineering and many other areas that deal with continuous problem domains.

***Example 3 (linguistic variable)*:** Consider the sentence "The amount of trading is heavy" which uses a fuzzy set "Heavy' to describe the quantity of the stock market trading in one day. More formally, this is expressed as:

Trading Quantity is Heavy

The variable *Trading Quantity,* which is a linguistic variable, in this example demonstrates an important concept in fuzzy logic.

(ii) **Universe:** Elements of a fuzzy set are taken from a *universe of discourse* or *universe* for short. The universe contains all elements that can come into consideration. Even the universe depends on the context, as the following example shows.

***Example 4 (Universe)*: (a)** The set of *young people* could have all human beings in the world as its universe. Alternatively it could be the numbers between 0 and 100; these would then represent age.
(b) The set $x > 10$ could have as a universe all positive measurements.

An application of the universe is to suppress faulty measurement data, for example negative values for the age in the above example. In case we are dealing with a non-numerical quantity, for instance *taste*, which cannot be measured against a numerical scale, we cannot use a numerical universe. The elements are then said to be taken from a *psychological continuum*; an example of such a universe could be {*bitter, sweet, sour, salt, hot, …*}.

(iii) **Membership Function:** Every element in the universe of discourse is a member of the fuzzy set to some grade, maybe even zero. The set of elements that have a non-zero membership is called the *support* of the fuzzy set. The function that

ties a number to each element x of the universe is called the *membership function* $\mu(x)$.

A fuzzy set is represented through its membership function. Depending on the nature of the fuzzy set it can be defined in two ways: (i) by enumerating membership values of those elements in the set (completely or partially), or (ii) by defining the membership function mathematically. Obviously, the first approach is possible only if the set is discrete, because a continuous fuzzy set has an infinite number of elements. Generally speaking, a fuzzy set A can be defined through enumeration using the expression given in Eqn.(1)

$$A = \sum_i \mu_A(x_i)|x_i \tag{1}$$

Where the summation operator refers to the union (disjunction) operation and the notation $\mu_A(x_i)|x_i$ refers to a fuzzy set containing exactly one (partial) element x_i with a membership value $\mu_A(x_i)$. For brevity, we do not list those elements x_i whose membership degree in set A is zero.

***Example 5 (fuzzy set through enumeration)*:** Let us consider a subset of natural numbers; say from 1 to 20, as the universe of discourse, U. We may define the fuzzy sets "small" and "medium" by enumeration as follows:

Small $\equiv 1/1 + 1/2 + 0.9/3 + 0.6/4 + 0.3/5 + 0.1/6$

Medium $\equiv 0.1/2 + 0.3/3 + 0.7/4 + 1/5 + 1/6 + 0.7/7 + 0.5/8 + 0.2/9$

***Example 6 (fuzzy set through defining membership function mathematically)*:** Alternatively, the fuzzy sets "small" and "medium" can be defined by explicitly describing their characteristics as shown in Eqns.(2) and (3) respectively.

$$\mu_{small}(x) = \begin{cases} 1 & x < 2 \\ \dfrac{7-x}{5} & 2 \le x \le 7 \\ 0 & x > 7 \end{cases} \tag{2}$$

$$\mu_{medium}(x) = \begin{cases} 0 & x < 1 \\ \dfrac{x-1}{4} & 1 \le x < 5 \\ 1 & 5 \le x < 6 \\ \dfrac{10-x}{4} & 6 \le x < 10 \\ 0 & x \ge 10 \end{cases} \tag{3}$$

Try the following exercises now.

E4) Let x be a linguistic variable that measures a university's academic excellence, which takes values from the universe of discourse $U = \{1,2,3,4,5,6,7,8,9,10\}$.

Suppose the term set of x includes *Excellent, Good, Fair,* and *Bad*. Express these fuzzy sets through enumeration.

E5) Order the fuzzy sets defined by the following membership grade functions (assuming $x \ge 0$) by the inclusion (subset) relation:

$$A(x) = \frac{1}{1+10x}, \quad B(x) = \left(\frac{1}{1+10x}\right)^{1/2}, \quad C(x) = \left(\frac{1}{1+10x}\right)^{1/2}$$

Now let us define the types of membership functions

Whereas there exist numerous types of membership functions, the most commonly used in practice are triangles, trapezoids, Gaussain, and Bell-shape functions. In the following these membership functions are briefly introduced:

(i) **Triangular Membership Function:** A triangular membership function is specified by three parameters $\{a,b,c\}$ as follows:

$$triangle(x:a,b,c)=\begin{cases}0 & x<a\\ \dfrac{x-a}{b-a} & a\le x\le b\\ \dfrac{c-x}{c-b} & b\le x\le c\\ 0 & x>c\end{cases} \quad (4)$$

The precise appearance of the function is determined by choice of parameters a, b, and c.

(ii) **Trapezoidal Membership Function:** A trapezoidal membership function is specified by four parameters $\{a,b,c,d\}$ as follows:

$$trapezoid(x:a,b,c,d)=\begin{cases}0 & x<a\\ \dfrac{x-a}{b-a} & a\le x<b\\ 1 & b\le x<c\\ \dfrac{d-x}{d-c} & c\le x\le d\\ 0 & x\ge d\end{cases} \quad (5)$$

The triangular membership function is a special case of the trapezoidal membership function. Due to their simple formulas and computational efficiency, both triangular and trapezoidal membership functions have proven popular with fuzzy logic practitioners and been used extensively, particularly in control.

(iii) **Gaussian Membership Function:** A Gaussian membership function is specified by two parameters $\{m,\sigma\}$ as follows:

$$Gaussian(x:m,\sigma)=e^{\left(\frac{(x-m)^2}{\sigma^2}\right)} \quad (6)$$

Where m and σ denote the centre and width of the function, respectively. We can control the shape of the function by adjusting the parameter σ. A small σ will generate a "thin" membership function, while a big σ will lead to a "flat" membership function.

(iv) **Bell-shaped Membership Function:** A Bell-shaped membership function is specified by three parameters $\{a,b,c\}$ as follows:

$$Bell(x:a,b,c)=\frac{1}{1+\left|\dfrac{x-c}{a}\right|^{26}} \quad (7)$$

where the parameter b is usually positive. A desired bell- shaped membership function can be obtained by a proper selection of the parameters a, b, and c. Specifically, we can adjust c and a to vary the center and width of the function and then use b to control the slopes at the crossover points.

Now, we are ready to design Membership Functions

One of the questions you may have by now is, "How do we determine the exact shape of the membership function for a fuzzy set?" What is important about membership function is that it provides gradual transition from regions completely outside a set to regions completely in the set.
A membership function can be designed in three ways:

(i) Interview those who are familiar with the underlying concept and later adjust it based on a tuning strategy.

(ii) Construct it automatically from data.

(iii) Learn it based on feedback from the system performance.

The first approach was the main approach used by fuzzy logic researchers and practitioners until the late 80s. Due to the lack of systematic strategies in those days, most fuzzy systems were tuned through trial-and-error process. This has become one of the main criticisms of fuzzy logic technology. Fortunately, many techniques in the second two categories have been developed since the late 80s using statistical techniques, neural networks, and genetic algorithms.

Even though one may attempt to define a membership function of arbitrary shape, it is generally recommended to use parametrerizable functions that can be defined by a small number of parameters. Using parameterized membership functions can not only reduce the system design time, it can also facilitate the automated tuning of the system because desired changes to the membership function can be directly related to corresponding changes in the related parameters.

The parameterizable membership function most commonly used in practice is the triangular membership function and the trapezoidal membership function. The former has three parameters and the latter has four parameters. Simplicity is the main advantage of triangular and trapezoidal membership functions.

To summarize we present the following guidelines for membership function design:

(i) Always use parameterizable membership functions. Do not define a membership function point by point.

(ii) Use a triangular or trapezoidal membership function, unless there is a good reason to do otherwise.

(iii) If you want to learn the membership function using neural network learning techniques, choose a differentiable (or even continuous differentiable) membership function (i.e., Gaussian).

So far, we discussed the fuzzy set with its membership function.

1.3 SUPPORT AND ALPHA-CUTS

In the following section, we shall discuss support and alpha-cuts.

The support of a fuzzy set A is the set of elements whose degree of membership in A is greater than 0. For example, the support of the fuzzy set in figure 3 is the open interval (10, 20). We can formally define support as follows:
Let A be a fuzzy set in the universe of discourse U. The support in A is defined as:

$$Spt(A) = \left\{ X \in U \,\middle|\, \mu_A(X) > 0 \right\} \tag{8}$$

Figure 3 – A fuzzy set with support set (10, 20)

The notion of $\alpha - \text{cut}$ (also called α-level) is more general than that of support. Let α_0 be a number between 0 and 1. The $\alpha - \text{cut}$ of a fuzzy set A at α_0, denoted as $A_{\alpha 0}$, is the set of elements whose degree of membership in A is no less than α_0. Mathematically, the $\alpha - \text{cut}$ of a fuzzy set A in U is defined as:

$$A_{\alpha 0} = \left\{ x \in U \,\middle|\, \mu_A(X) \geq \alpha_0 \right\} \tag{9}$$

Figure 4 – The fuzzy membership function of *moderately approved* presidential candidate

***Example 7 (α-cut)*:** Let us consider the concept "moderately approved" regarding the public's opinion of a presidential candidate. The universe of discourse is the percentage of those people supporting the candidate in a national poll. To simplify our discussion we use a discrete universe of discourse $U = \{0\%, 10\%, 20\%, \cdots, 100\%\}$. The membership function of the moderately fuzzy set approved is shown in figure 1.4. We can construct the $\alpha - \text{cut}$ of the fuzzy set at 0.7 as:
Moderately Approved$_{0.7}$ = {40%, 50%, 60%}
Similarly, we can construct the following $\alpha - \text{cut}$:
Moderately Approved$_{0.2}$ = {30%, 40%, 50%, 60%, 70%, 80%}
Moderately Approved$_{0.6}$ = {40%, 50%, 60%, 70%}
Obviously, as the α value increases, the set generated by α-cutting, the fuzzy set becomes smaller.

Now, try these exercises.

E6) Construct the $\alpha - \text{cut}$ at $\alpha = 0$ and $\alpha = 1$ for the fuzzy set moderately approved shown in figure 1.3.

E7) Construct the $\alpha - \text{cut}$ at $\alpha = 0.4$ for the fuzzy sets defined in exercise E4.

E8) If $\alpha_1 < \alpha_2$ then prove that $A_{\alpha 1} \supseteq A_{\alpha 2}$, where $\supseteq$ denotes a crisp superset (i.e., suppositions) relation.

So far, we discussed classical set, fuzzy set, support and α-cut. In the following section, we shall discuss hedges.

1.4 HEDGES

Hedges are also called fuzzy set transformers and are used to model adjectives or adverbs, whose role in natural language is to modify the semantics of an adjective or noun on which they operate. Hedges are used to intensify or dilute the membership function of a fuzzy set. The shape of a fuzzy set can be changed through the use of contrasts and restrictions. When operating on fuzzy numbers, hedges can increase or decrease the expectancy of a fuzzy number. On the basis of underlying actions the following categories of hedges are defined in literature.

- ***Approximation hedges*** – Such hedges are used to increase or decrease the expectancy of a fuzzy number. Some of the linguistic qualifiers modeled using approximation hedges are *about, around, near, close to,* and *in vicinity of.* The approximation hedges allow fuzzy models to treat all numeric values as fuzzy numbers, thus reducing complexity and allowing a uniform approach to knowledge acquisition and management.

- ***Contrast hedges –*** Such hedges are used to increase or decrease the membership function's central measure of membership by focusing in the area around 50% membership space. Example hedges of this category are *almost, definitely, positively, generally,* and *usually*.
- ***Dilution hedges –*** Hedges belonging to this category softens the membership function over the fuzzy set. Dilution hedges include *quite, rather, slightly,* and *somewhat*.
- ***Intensification hedges –*** Hedges belonging to this category hardens the membership function over the fuzzy set. Intensification hedges include *very,* and *extremely*.
- ***Negation hedges*** – Negation hedges reverse the truth membership of the fuzzy set. "Not" is a primary negation hedge that produces the complement of a fuzzy set.
- ***Restriction hedges*** – Hedges belonging to this category restricts the membership function relative to the shape of the underlying fuzzy set. Restriction hedges include above, below, more than, and less than.

Most hedges involve complex algorithmic processing of the membership function. The basic intensification and dilution hedges, however, simply modify each point on the fuzzy set by applying a power function. Eqn.(10) shows the power function for the intensification hedge.

$$\mu_{intesify(A)}(x_i) = \mu_A^n(x_i) \tag{10}$$

The "very" hedge has n = 2 and so simply squares the membership value; "slightly" has n = 1.2 and other intensification hedges have stronger or weaker exponents on the membership function. Eqn.(11) shows the power function for the dilution hedge.

$$\mu_{dilute(A)}(x_i) = \mu_A^{1/n}(x_i) \tag{11}$$

The "somewhat" hedge has n = 2 and so simply takes the square root of the membership values. Other forms of the dilution hedge class supply different exponent values and hence take different roots.

Hedges are used in fuzzy models to dynamically create new fuzzy sets and change the meaning of linguistic variables. This enables the modification of existing fuzzy sets temporarily to provide different meaning to the underlying linguistic variable. The modified shapes of a fuzzy number "*around 50,000*" after applying "*somewhat*" and "*very*" hedges on it are shown in figure 5.

Figure 5 – A fuzzy number "around 50,000" and its modified shape after applying "somewhat" and "very" hedges.

Try an exercise.

E9) Apply the "very" hedge on the fuzzy sets defined in exercise E4 to get the new modified fuzzy sets. Sow the modified fuzzy sets through numeration.

In this section, we shall discuss various operations on fuzzy sets.

1.5 OPERATIONS ON FUZZY SETS

The three fundamental operations in classical sets are union, intersection, and complement. The union of two sets A and b (denoted as $A \cup B$) is the collection of those objects that belongs to either A or B. The intersection of A and B (denoted as $A \cap B$) is the collection of those objects that belong to both A and B. The complement of a set A (denoted as A^c or $\overline{A}$) is the collection of objects not belonging to A. Like classical sets, the operations, union, intersection, and complement can also be defined for the fuzzy sets. The membership function is obviously a crucial component of a fuzzy set. It is therefore natural to define operations on fuzzy sets by means of their membership functions. Since membership in a fuzzy set is a matter of degree, set operations should be generalized accordingly. The fuzzy intersection operation turns out to be mathematically equivalent to the fuzzy conjunction (AND) operation, because they have identical desired properties. Similarly, the fuzzy union operation is mathematically equivalent to the fuzzy disjunction (OR) operation and the fuzzy complement is equivalent to the fuzzy negation (NOT) operation.

1.3.1 The Fuzzy AND operator

Let A and B be two fuzzy sets on mutual universe of discourse with membership function μ_A and μ_B. The intersection of fuzzy sets A and B is defined as:

$$\mu_{A \cap B}(x) = \min\left(\mu_A(x), \mu_B(x)\right) \quad (12)$$

Figure 6 illustrates the fuzzy region (shown in bold black color) produced by the proposition *Young Adult And Middle-Aged*. This is the region where an age is member of (compatible with) both the *young adult middle-aged* concepts.

Figure 6 – The fuzzy AND operator

1.3.2 The Fuzzy OR Operator

Let A and B be two fuzzy sets on mutual universe of discourse with membership functions μ_A and μ_B. The intersection of fuzzy sets A and B is defined as: $\mu_{A \cup B}(x) = \max\left(\mu_A(x), \mu_B(x)\right)$ (13)

Figure 1.5 illustrates the fuzzy region produced by the proposition *Young adult Or Middle-Aged*. This is the region where an age is member of (compatible with) either the *young adult* or *middle-aged* concepts. In figure 7, the bold line traces the resulting truth function. The area represented by the twin peaks, spans the width of both fuzzy sets.

Figure 7 – The fuzzy OR operator

1.3.3 The Fuzzy NOT Operator

Let A be a fuzzy set defined on a universe of discourse U with membership function μ_A. The negation of the fuzzy set A is defined as:

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x) \quad (14)$$

Figure 8 illustrates the fuzzy region produced by the proposition *Not Middle-aged*. This is the region in the linguistics variables supporting Client age that forms the complement of the middle-aged concept. In figure 6, the bold line traces the resulting truth function.

Figure 8 – The fuzzy NOT operator

***Example 8 (Fuzzy operators)*:** A four person family wants to buy a house. An indication of how comfortable they are to be is the number of bedrooms in the house. But they also want a large house. Let $U = \{1,2,3,4,5,6,7,8,9,10\}$ be the set of available houses described by their number of bedrooms. Then the fuzzy set C (for comfortable) may be described as

$$C = \left\{\frac{0.2}{1}, \frac{0.5}{2}, \frac{0.8}{3}, \frac{1.0}{4}, \frac{0.7}{5}, \frac{0.3}{6}, \frac{0}{7}, \frac{0}{8}, \frac{0}{9}, \frac{0}{10}\right\}$$

Let L be the fuzzy set *Large* defined as $L = \left\{\frac{0}{1}, \frac{0}{2}, \frac{0.2}{3}, \frac{0.4}{4}, \frac{0.6}{5}, \frac{0.8}{6}, \frac{1}{2}, \frac{1}{8}, \frac{1}{9}, \frac{1}{10}\right\}$

Then, $C \cap L = \left\{\frac{0}{1}, \frac{0}{2}, \frac{0.2}{3}, \frac{0.4}{4}, \frac{0.6}{5}, \frac{0.3}{6}, \frac{0}{7}, \frac{0}{8}, \frac{0}{9}, \frac{0}{10}\right\}$

To interpret this, five bedrooms is optimal, but only satisfactory to the grade 0.6. The second best solution is four bedrooms.

The union of the fuzzy sets *Comfortable* and *Large* is

$$C \cup L = \left\{\frac{0.2}{1}, \frac{0.5}{2}, \frac{0.8}{3}, \frac{1}{4}, \frac{0.7}{5}, \frac{0.8}{6}, \frac{1}{7}, \frac{1}{8}, \frac{1}{9}, \frac{1}{10}\right\}$$

Here four bedrooms is fully satisfactory (1,0) because it is comfortable, and 7-10 bedrooms also, because that would mean a large house.

The complement of Large is

$$\bar{L} = \left\{\frac{1}{1}, \frac{1}{2}, \frac{0.8}{3}, \frac{0.6}{4}, \frac{0.4}{5}, \frac{0.2}{6}, \frac{0}{7}, \frac{0}{8}, \frac{0}{9}, \frac{0}{10}\right\}$$

Now you can try the following exercises.

E10) Let A and B are two fuzzy sets and $x \in U$, If $\mu_A(x) = 0.3$ and $\mu_B(x) = 0.9$ then find out the following membership values:
(i) $\mu_{\bar{A} \cup B}(x)$ (ii) $\mu_{A \cap B}(x)$ (iii) $\mu_{\bar{A} \cup \bar{B}}(x)$ (iv) $\mu_{\bar{A} \cap \bar{B}}(x)$ (v) $\mu_{\overline{A \cup B}}(x)$ (vi) $\mu_{\overline{A \cap B}}(x)$

E11) For the above example "buying a house" what will be the elements of the following fuzzy sets
(i) $\overline{C \cup L} = \bar{C} \cap \bar{L}$ (ii) $\overline{C \cap L} = \bar{C} \cup \bar{L}$

E12) By using the membership function defined in exercise E4 construct the membership functions of the following compound sets

(i) Not Bad but Not Very Good

(ii) Good but Not Excellent

In the following section, we shall discuss the fuzzy relation.

1.6 FUZZY RELATION

A fuzzy relation generalizes the notion of a classical black-and-white relation into one that allows partial membership. For example a binary relation "friend" will classify all human relationship into either being friend or not being friend. A fuzzy relation "friend", in contrast, can describe the degree of friendship between two persons.

The classical notion of relation describes a relationship that holds between two or more objects. A relationship between two objects is represented by a *binary relation*, a relation with two arguments. For example, the parent relationship between a parent and his/her child can be represented as a binary relation. More generally, we can use an n-ary relation, a relation with n arguments, to describe a relationship between n objects. For instance, we can use an n-ary relation to describe that student X took course Y during semester Z in year W. This relation has four arguments such as took_course(student, course, semester, year). An n-ary relation can be formally defined as set of ordered listed of n objects. Each list describes a case in which the relation holds.

A binary relation on variables x and y, whose domains are X and Y respectively, can be defined as a set of ordered pairs in $X \times Y$. For instance, the binary relation "less than" between two real numbers can be formally defined as
$R = \{(x, y) | x < y; x, y \in R\}$

It is easy to see that the relation is a subset of $X \times Y$. In general, an n-ary relation on $x_1, x_2, \cdots, x_n$ whose domains are $X_1, X_2, \cdots, X_n$ is a subset of $X_1 \times X_2 \times \cdots \times X_n$.

Since a relation can be viewed as a set, we can easily generalize the classical notion of relation using fuzzy sets. We will show how to generalize binary relations. First we can represent a binary relation R on x, y with domains X, Y as a function that maps an ordered pair (x, y) in $X \times Y$ to 0 (i.e., $R = X \times Y \rightarrow \{0,1\}$.

A fuzzy relation generalizes the classical notion of relation into a matter of degree. As mentioned earlier, the fuzzy relation *friend* describes the degree of friendship between two persons. Similarly, a fuzzy relation *Petite* between height and weight of a person describes the degree by which a person with a specific height and weight is considered petite. Formally, fuzzy relation R between variables x and y, whose domains are X and Y, respectively, is defined by a function that maps ordered pairs in $X \times Y$ to their degree in the relation, which is a number between 0 and 1, i.e., $R = X \times Y \rightarrow [0,1]$.

More generally, a fuzzy n-ary relation R in $x_1, x_2, \cdots, x_n$ whose domains are $X_1 \times X_2 \times \cdots \times X_n$, respectively, is defined by a function that maps an n-ary $(x_1, x_2, \cdots, x_n)$ in $X_1 \times X_2 \times \cdots \times X_n$ to a number in the interval [0, 1]. i.e., $R : X_1 \times X_2 \times \cdots \times X_n \rightarrow [0,1]$. Just as a classical relation can be viewed as a set, a fuzzy relation can be viewed as a fuzzy subset. From this perspective, the mapping above is equivalent to the membership function of a multidimensional fuzzy set.

If the possible values of x and y are discrete, we can express a fuzzy relation in a matrix form. For example, suppose we wish to express a fuzzy relation Petite in terms of the height and the weight of a female. Suppose the range of the height and the weight of interest to us are {5',5'1", 5'2", 5'3", 5'4", 5'5", 5'6"}, denoted h, and {90, 95, 100, 105, 110, 115, 120, 125} (in lb,) denoted w, respectively. We can express the fuzzy relation in a matrix form as shown in table 1.

Table 1 A fuzzy relation – "Petite"

	90	95	100	105	110	115	120	125
5	1	1	1	1	1	1	0.5	0.2
5′1″	1	1	1	1	1	0.9	0.3	0.1
5′2″	1	1	1	1	1	0.7	0.1	0
5′3″	1	1	1	1	0.5	0.3	0	0
5′4″	0.8	0.6	0.4	0.2	0	0	0	0
5′5″	0.6	0.4	0.2	0	0	0	0	0
5′6″	0	0	0	0	0	0	0	0

Each entry in the matrix indicates the degree a female with the corresponding height (i.e., the row heading) and weight (i.e., the column heading) is considered to be *petite*. For instance, the entry corresponding to a height of 5'3" and a weight of 115 lb. has a value 0.3, which is the degree to which such a female person will be considered a *petite* person; i.e., petite(5'3", 115 lb) =0.3.

Once we define the Petite fuzzy relation, we can answer two kinds of questions:

- What is the degree that a female with a specific height and a specific weight considered to be petite?
- What is the possibility that a petite person has a specific pair of height and weight measures?

In answering the first question, the fuzzy relation is equivalent to the membership function of a multidimensional fuzzy set. In the second case, the fuzzy relation becomes a possibility distribution assigned to a petite whose actual height and weight are unknown.

Let us now summarize the unit.

1.7 SUMMARY

In this unit, we have discussed the fundamental concepts of the fuzzy set and fuzzy relation. Specifically we have covered the following:

1. Elaborated the essence of fuzzy set to solve real life complex problems.
2. Provided a comparative introduction of the fuzzy set vs. classical set.
3. Defined membership function and its role to represent fuzzy set.
4. Provided details about well-known membership functions along with guidelines to define a membership function.
5. Introduced support and alpha-cut of fuzzy sets.
6. Defined hedges that are used to change the shapes of the fuzzy sets.
7. Introduced basic set theoretic operations that are also applicable on fuzzy sets with examples.
8. Defined fuzzy relation as a generalization of the classical relations.

1.8 SOLUTIONS/ANSWERS

E1) A= {3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61,67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149}
B = {3, 5}
C = {5, 7, 13, 19, 31, 43, 61, 73, 103, 109, 139}

E4) Excellent = 0.2/8+0.6/9+1/10

Good = 0.1/6+0.5/7+0.9/8+1/9+1/10

Fair = 0.3/2+0.6/3+0.9/4+1/5+0.9/6+0.5/7+0.1/8

Bad = 1/1+0.7/2+0.4/3+0.1/4

E5) $C(x)=\left(\frac{1}{1+10x}\right)^2$, $A(x)=\frac{1}{1+10x}$, $B(x)=\left(\frac{1}{1+10x}\right)^{1/2}$

E6) Moderately Approved $_{0.0}$ = {30%, 40%, 50%, 60%, 70%, 80%}

Moderately Approved $_{0.2}$ = {50%, 60%}

E7) Excellent $_{0.4}$ = {9, 10}

Good $_{0.4}$ = {7, 8, 9, 10}

Fair $_{0.4}$ = {3, 4, 5, 6, 7, 8}

Bad $_{0.4}$ = {1, 2, 3}

E8) Given that $\alpha_1 < \alpha_2$. We have to prove that $A_{\alpha 1} \supseteq A_{\alpha 2}$.

Let $x \in A_{\alpha 2}$. By definition to alpha-cut we have

$\mu_A(x) \geq \alpha_1$,

$\Rightarrow \mu_A(x) \geq \alpha_1$, since $\alpha_1 < \alpha_2$

$\Rightarrow x \in A_{\alpha 1}$

Hence $A_{\alpha 1} \supseteq A_{\alpha 2}$.

E9) Very Excellent = 0.04/8+0.36/9+1/10

Very Good = 0.01/6+0.25/7+0.81/8+1/9+1/10

Very Fair = 0.09/2+0.36/3+0.81/4+1/5+0.81/6+0.25/7+0.01/8

Very Bad = 1/1+0.49.2/2+0.16/3+0.01/4

E10) (i) $\mu_{A\cup B}(x)=0.9$

(ii) $\mu_{A\cap B}(x)=0.3$

(iii) $\mu_{\bar{A}\cup\bar{B}}(x)=0.6$

(iv) $\mu_{\bar{A}\cap\bar{B}}(x)=0.1$

(v) $\mu_{\overline{A\cup B}}(x)=0.1$

(vi) $\mu_{\overline{A\cap B}}(x)=0.6$

E11) (i) $\overline{\overline{C\cup L}}=\{0.8,0.5,0.2,0,0.3,0.2,0,0,0,0\}$.....(a)

$\bar{C}\cap\bar{L}=\{0.8,0.5,0.2,0,0.3,0.7,1,1,1,1\}\cap\{1,1,0.8,0.6,0.4,0.2,0,0,0,0\}$

$=\{0.8,0.5,0.2,0,0.3,0.2,0,0,0,0\}$.....(b)

From (a) and (b) we have $\overline{\overline{C\cup L}}=\bar{C}\cap\bar{L}$

(ii) $\overline{\overline{C\cup L}}=\{1,10.8,0.6,0.4,0.7,1,1,1,1,\}$.....(a)

$\bar{C}\cap\bar{L}=\{0.8,0.5,0.2,0,0.3,0.7,1,1,1,1\}\cap\{1,1,0.8,0.6,0.4,0.2,0,0,0,0\}$

$=\{1,1,0.8,0.6,0.4,,0.7,1,1,1,1,\}$.....(b)

From (a) and (b) we have $\overline{\overline{C\cup L}}=\bar{C}\cap\bar{L}$

E12) (i) From the answer of E4 we have,
Bad = 1/1+0.7/2+0.4/3+0.1/4
Therefore, Not Bad = 0.3/2+0.6/3+0.9/4+1/5+1/6+1/7+1/8+1/9+1/10....(a)
Similarly, we have from E4,

Good = 0.1/6+0.5/7+0.9/8+1/9+1/10
Therefore, Very Good = 0.01/6+0.25/7+0.81/8+1/9+1/10
Not Very Good = 1/1+1/2+1/3+1/4+1/5+0.99/6+0.75/7+0.19/8…..(b)
Apply the fuzzy intersection between (a) and (b) we get,
Not Bad but Not Very Good = 0.3/2+0.6/3+0.9/4+1/5+0.99/6 0.75/70. 19/8

(ii) From E4 we have
Good = 0.1/6+0.5/7+0.9/8+1/9+1/10 … (a)
Excellent = 0.2/8+0.6/9+1/10 ... (b)
Applying fuzzy NOT operation on (a) we get,
Not Excellent = 1/1+1/2+1/3+1/4+1/5+1/6+1/7+0.8/8+0.4/9 … (c)
Now, applying fuzzy intersection operation between (a) and (c) we get,
Good but Not Excellent = 0.1/6+0.5/7+0.8/8+0.4/9

UNIT 2 FUZZY CLUSTERING

Structure **Page No**

2.1 INTRODUCTION

Clustering is an important data analysis tool that provides improved data understanding. It plays a key role in searching for structures in data. Cluster analysis is aimed at dividing data whose identity is not known in advance, into homogeneous groups or clusters such that similar data objects belong to the same cluster and dissimilar data objects to different clusters. A *cluster* is, therefore, a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. In other words, given a finite set of data, X, the problem of clustering in X is to find several cluster centers that are required to form a *partition* of X such that the degree of association is strong for data within blocks of the partition and weak for data in different blocks. However, this requirement is too strong in many practical applications, and it is thus desirable to replace it with a weaker requirement. When the requirement of a crisp partition of X is replaced with a weaker requirement of a *fuzzy partition* or a *fuzzy pseudo partition* on X, we refer to the emerging problem area as *fuzzy clustering* as in Section 2.4. Fuzzy pseudo partitions are often called *fuzzy c-partitions*, where c designates the number of fuzzy classes in the partition.

Clustering is an active research area and there are wide arrays of clustering algorithms that have been proposed in literature are discussed in Section 2.6. These algorithms differ from each other in their approach, in the cluster quality they achieve and also in terms of computational efficiency. In Section 2.7, we will look at these issues in the light of fuzzy clustering techniques. There are two basic methods of fuzzy clustering. One of them, which are based on fuzzy c-partitions, is called a *fuzzy equivalence relation-based hierarchical clustering method*. We will see these methods in details in this unit. Various modifications of these methods can be found in the literature. We have also introduced some major application areas of fuzzy clustering in Section 2.8.

Objectives

After studying this unit, you should be able to

- know the essence the concept of clustering;
- define the fuzzy clustering;
- differentiate the classical cluster analysis and the fuzzy cluster analysis;
- know various distance function useful for clustering;
- apply fuzzy clustering to solve real-life problems.

2.2 WHAT IS CLUSTERING?

Data clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Clustering is the classification of similar objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait – often proximity according to some defined distance measure. Machine learning typically regards data clustering as a form of unsupervised learning. Besides the term *data clustering* (or just *clustering*), there are a number of term with similar meanings, including *cluster analysis*, *automatic classification*, *numerical taxonomy*, *botryology* and *typological analysis*.

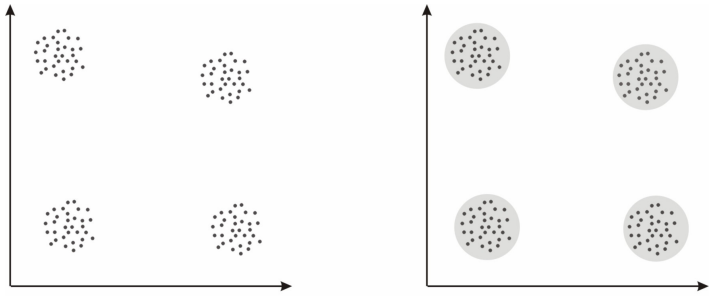

Fig. 1: An example of clusters.

Clustering deals with finding a *structure* in a collection of unlabeled data. A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way". A *cluster* is, therefore, a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. We can show this with a simple graphical example shown in Fig.1.

In this case we easily identify the 4 clusters into which the data can be divided where the similarity criterion is *distance*, i.e., two or more objects belong to the same cluster if they are "close' according to a given distance (in this case geometrical distance). This is called *distance-based clustering*. Another kind of clustering is *conceptual clustering* in which two or more objects belong to the same cluster if it defines a concept *common* to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures.

The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute "best" criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs.

For instance, we could be interested in finding representatives for homogeneous groups (*data reduction*), in finding "natural clusters" and describe their unknown properties (*"natural" data types*), in finding useful and suitable groupings (*"useful data classes*) or in finding unusual data objects (*outlier detection*).

Clustering techniques have been extensively studied in statistics, machine learning and data mining. Clustering has many interesting commercial applications, and is extensively applied for studying the data pertaining to an organisational application. It helps in discovering hitherto unknown patterns in data, which can be fruitfully engaged in taking future marketing decisions. Some common uses of clustering are stated here:

- **Computational Biology and Bioinformatics** – In transcriptomics, clustering is used to build groups of genes with related expression patterns. Often such groups contain functionally related proteins, such as enzymes for a specific pathway, or genes that are co-regulated. High throughput experiments using expressed sequence tags (ESTs) or DNA microarrays can be a powerful tool for genome annotation, a general aspect of genomics.

  In sequence analysis, clustering is used to group homologous sequences into gene families. This is a very important concept in bioinformatics, and evolutionary biology in general. See evolution by gene duplication.

- **Plant and Animal Ecology** – In the fields of plant and animal ecology, clustering is used to describe and to make spatial and temporal comparisons of communities (assemblages) of organisms in heterogeneous environments; it is also used in plant systematic to generate artificial phylogenies or clusters of organisms (individuals) at the species, genus or higher level that share a number of attributes.

- **Marketing Research** – Cluster analysis is widely used in market research when working with multivariate data from surveys and test panels. Market researchers use cluster analysis to partition the general population of consumers into market segments and to better understand the relationship between different groups of consumers/potential customers for the following purposes.

  - Segmenting the Market and determining target markets
  - Product positioning
  - New Product development
  - Selecting test markets

- **Insurance** – Insurance companies use clustering to identify policy holders with high average policy claims; identifying frauds.

- **City-Planning** – Identifying groups of houses according to their house type, value and geographical location.

- **Earthquake Studies** – Past data on earthquake is clustered to study the nature of continental faults.

- **Document Clustering –** Clustering proves to be one of the most promising techniques to reign in the huge unwieldy collection of documents on electronic media. Clustering similar documents together and representing them with a cluster center helps users get a glimpse of contents of a whole collection together without the tedious task of reading all of them.

- **Data Mining** – Many data mining applications involve partitioning data items into related subsets; the marketing applications discussed above represent some examples. Another common application is the division of documents, such as World Wide Web pages, into genres.

- **Social Network Analysis** – In the study of social network, clustering may be used to recognize communities within large groups of people.

- **Image Segmentation** – Clustering can be used to divide a digital image into distinct regions for border detection or object recognition.

- **WWW** – Document classification; clustering web-log data to discover groups of similar access patterns.

The main requirements that a clustering algorithm should satisfy are:

- Scalability
- Dealing with different types of attributes

- Discovering clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Ability to deal with noise and outliers
- Insensitivity to order of input records
- High dimensionality
- Interpretability and usability

There are a number of limitations with clustering. Among them:

- current clustering techniques do not address all the requirements adequately (and concurrently);
- dealing with large number of dimensions and large number of data items can be problematic because of time complexity;
- the effectiveness of the method depends on the definition of "distance" (for distance-based clustering);
- if and *obvious* distance measure doesn't exist we must "define" it, which is not always easy, especially in multi-dimensional spaces;
- the result of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways.

Now we shall discuss fuzzy cluster analysis in the following sections.

2.3 FUZZY CLUSTER ANALYSIS

Cluster analysis divides data into groups (clusters) such that similar data objects belong to the same cluster and dissimilar data objects to different clusters. The resulting data partition improves data understanding and reveals its internal structure. Partitional clustering algorithms divide up a data set into clusters or classes, where similar data objects are assigned to the same cluster whereas dissimilar data objects should belong to different clusters. In real applications there is very often no sharp boundary between clusters so that fuzzy clustering is often better suited for the data. Membership degree between zero and one are used in fuzzy clustering instead of crisp assignments of the data to clusters. The most prominent fuzzy clustering algorithm is the fuzzy c-means, a fuzzification of k-means.

Areas of application of fuzzy cluster analysis include for example data analysis, pattern recognition, and image segmentation. The detection of special geometrical shapes like circles and ellipses can be achieved by so-called shell clustering algorithms. Fuzzy clustering belongs to the group of soft computing techniques (which include neural nets, fuzzy systems, and genetic algorithms).

2.4 CLASSICAL VS. FUZZY CLUSTERING

Clustering is motivated by the need to find interesting patterns or groupings in a given set of data. For example, a market analysis company may want to collect data about a group of customers (e.g., through a survey questionnaire or interviews) and analyze these data by finding interesting groupings of customers. The result of such an analysis can be used by a company to develop its marketing strategy or to develop a family of product lines that are targeted toward various customer groupings.

In the area of pattern recognition and image processing, clustering is often used to perform the task of "segmenting" the image (i.e., partitioning pixels on an image into regions that correspond to different objects or different faces of objects in the images).

This is because image segmentation can be viewed as a kind of data clustering problem where each datum is described by a set of image features (e.g., intensity, color, texture, etc.) of a pixel.

Classical clustering approaches generate partitions in a partition; each pattern belongs to only one cluster. So, the clusters in a partition are disjoint. In other words, classical clustering algorithms find a "hard partition" of a given dataset based on certain criteria that evaluate the goodness of a partition. By "hard partition" we mean that each datum belongs to exactly one cluster of the partition. More formally, we can define the concept "hard partition" as follows:

Definition 1 (Hard Partition): Let X be a set of data, and x_1 be an element of X. A partition $P = \{C_1, C_2, \ldots, C_k\}$ of X is "hard" if and only if the following two condition hold.

(i) for all $x_i \in X$ there exists a partition $C_i \in P$ such that $x_i \in C_j$.

(ii) for all $x_i \in X$, $x_i \in C_j \Rightarrow x_i \notin C_k$ where $k \neq j$, $C_k C_j \in P$.

The first condition in the definition assures that the partition covers all data points in X, whereas the second condition assures that all clusters in the partition are mutually exclusive.

In many real-world clustering problems, however, some data points partially belong to multiple clusters, rather than to a single cluster exclusively. For example, a pixel in a Magnetic Resonance Image (MRI) may correspond to a mixture of two different types of tissues. A particular customer may be a "borderline case" between two groups of customers (e.g., between moderate conservatives and moderate liberals). These observations motivated the development of the "fuzzy clustering" algorithm. A fuzzy algorithm finds "soft partition" of a given data set based on certain criteria. In a soft partition, a datum can partially belong to multiple clusters.

Formally this can be defined as follows:

Definition 2 (Soft Partition): Let X be a set of data, and x_i be an element of X. A partition $P = \{C_1, C_2, \ldots, C_k\}$ of X is "soft" if and only if the following two conditions hold.

(i) for all $x_i \in X$ and for all $C_j \in P$, $0 \leq \mu_{cj}(x_i) \leq 1$ such that $x_i \in C_j$.

(ii) for all $x_i \in X$ there exists $C_j \in P$ such that $\mu_{cj}(x_i) > 0$.

where $\mu_{cj}(x_i)$ denote clustering of special interest to which x_i belongs to cluster C_j.

A type of fuzzy clustering of special interest is one that ensures the membership degree of a point x in all clusters adding up to one, i.e.,

$$\sum_j \mu_{cj}(x_i) = 1 \qquad \forall x_i \in X$$

A soft partition that satisfies this additional condition is called a *constrained soft partition*. The fuzzy c-means algorithm, which is best known fuzzy clustering algorithm, produces a constrained partition.

2.5 DISTANCE MEASURE IN CLUSTERING

The central idea of clustering is to group data points in such a way so that similar elements belong to the same cluster and objects belonging to different clusters are

more different from each other than those belonging to the same cluster. To formalize the idea of *closeness* or *nearness*, central to clustering is the concept of a *distance measure* between data points. If the components of the data instance vectors are all in the same physical units then it is possible that the simple Euclidean distance metric is sufficient to successfully group similar data instances. However, even in this case the Euclidean distance can sometimes be misleading. Fig. 2 illustrates this with an example of the width and height measurements of an object. Despite both measurements being taken in the same physical units, an informed decision has to be made as to the relative scaling. As Fig. 2 shows, different scaling can lead to different clustering.

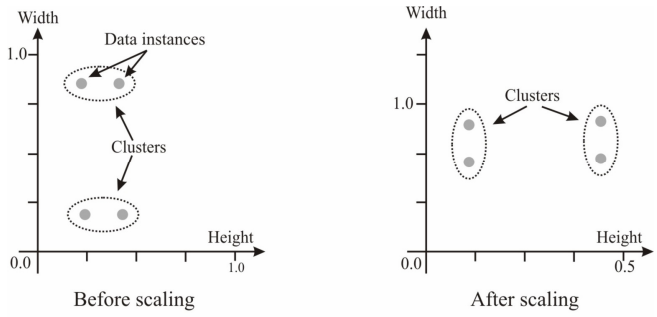

Fig. 2: Clusters before scaling and after scaling.

Notice however that this is not only a graphic issue: the problem arises from the mathematical formula used to combine the distances between the single components of the data feature vectors into a unique distance measure that can be used for clustering purposes: different formulas leads to different clustering. Again, domain knowledge must be used to guide the formulation of a suitable distance measure for each particular application.

For higher dimensional dart, a popular measure is the Minkowski metric,

$$d_p(x, y) = \left(\sum_{i=1}^{d} | x_i - y_i |^p \right)^{1/p}$$

where d is the dimensionality of the data. The *Euclidean* distance is a special case where $p = 2$, while *Manhattan* metric has $p = 2$. However, there are no general theoretical guidelines for selecting a measure for any given application. It is often the case that the components of the data feature vectors are not immediately comparable. It can be that the components are not continuous variables, like length, but nominal categories, such as the days of the week. In these cases again, domain knowledge must be used to formulate an appropriate measure.

Suppose x and y are two data points in a dataset and $d(x, y)$ denotes the distance between x and y and satisfies the following properties:

(i) $d(x, x) = 0$ i.e., a data item is at distance zero from itself.

(ii) $d(x, y) = d(y, x)$ i.e., distance is a symmetric notion.

(iii) $d(x, y) \leq d(x, y) + d(z, y)$ i.e., distance measure satisfies the triangle inequality.

Given a k-dimensional data set, the data item x and y can be represented as $x = (x_1, x_2, \ldots, x_k)$ and $y = (y_1, y_2, \ldots, y_k)$. Some of the commonly used distance measures for clustering can be summarized as follows:

(i) Euclidean distance or "L_2 norm" which is given by

$$d(x, y) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

(ii) Manhattan distance or "L_1 norm" which is given by

$$d(x, y) = \sum_{i=1}^{k} | x_i - y_i |$$

(iii) Maximum of dimensions or L_∞ norm which is given by

$$\max_{i=1}^{k} | x_i - y_i |$$

(iv) Cosine measure – This is a non-Euclidean distance measure and represents a similarity measure between two patterns. If the two patterns are represented as two vectors, the cosine of angle between the two vectors is identical to their correlation coefficient. The pattern similarity measure is thus given by

$$\text{similarity } (x, y) = \frac{\sum (xy)}{\sum x^2 \sum y^2}$$

(v) Minkowski metric – This is given by

$$d_p(x, y) = \left(\sum_{i=1}^{d} | x_i - y_i |^p \right)^{1/p}$$

where, d is the dimensionality of the data.

In the following section, we shall discuss clustering algorithm.

2.6 CLUSTERING ALGORITHMS

Clustering algorithms may be classified as listed below:

- **Exclusive Clustering** – In this approach data are grouped in an exclusive way, so that if a certain datum belongs to a definite cluster then it could not be included in another cluster. A simple example of it is shown in the Fig. 3, where the separation of points is achieved by a straight line on a bi-dimensional plane. K-means algorithm is an example of exclusive clustering.

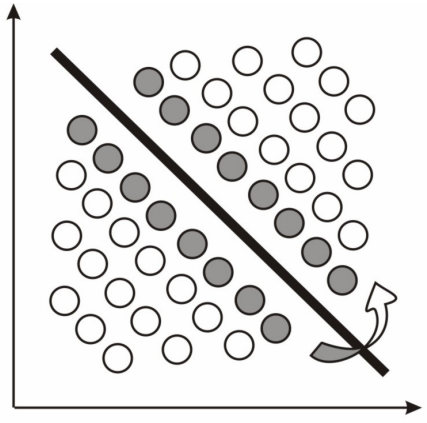

Fig. 3: Clusters before scaling and after scaling.

- **Overlapping Clustering –** The overlapping clustering, uses fuzzy sets to cluster data, so that each point may belongs to two or more clusters with different degrees of membership. In this case, data will be associated to an appropriate membership value. Fuzzy c-means is an overlapping clustering.

- **Hierarchical Clustering –** Instead, a hierarchical clustering algorithm is based on the union between the two nearest clusters. The beginning condition is realized by setting every datum as a cluster. After a few iterations it reaches the final clusters wanted.
- **Probabilistic Clustering** – Such type of clustering use a completely probabilistic approach. Mixture of Gaussian is an example of probabilistic clustering.

2.7 FUZZY CLUSTERING ALGORITHMS

As discussed earlier, traditional clustering approaches generate partitions; in a partition, each pattern belongs to only one cluster. So, the clusters in a partition are disjoint. Fuzzy clustering extends this notion to associate each partition with every cluster using a membership function. The output of such algorithms is a clustering, but not a partition. Given a set data $X = \{x_1, x_2, \ldots, x_n\}$, where x_k , in general, is a vector

$$x_k[x_{k1}, x_{k2}, \ldots, x_{kp}] \in \Re^p$$

For all $k \in N_n$ the problem of fuzzy clustering is to find a fuzzy pseudo partition and the associated cluster centers by which the structure of the data is represented as best as possible.

Definition 3 (Fuzzy Pseudo Partition or Fuzzy c-Partition): Let $x = \{x_1, x_2, \ldots, x_n\}$ be a set of given data. A fuzzy pseudo partition or fuzzy c-partition of X is a family of fuzzy subsets of X, denoted by $P\{A_1, A_2, \ldots, A_n\}$ which satisfies

$$\sum_{i=1}^{c} A_1(x_k) = 1$$

for all $k \in N_n$ and

$$0 < \sum_{k=1}^{n} A_i(x_k) < n$$

for all $i \in N_c$, where c is a positive integer.

Example 1: Let $X = \{x_1, x_2, x_3\}$ and

$$A_1 = 0.6/x_1 + 1/x_2 + 0.1/x_3$$

$$A_2 = 0.4/x_1 + 0/x_2 + 0.9/x_3$$

Then $\{A_1, A_2\}$ is a pseudo partition or fuzzy 2-partition of X. Fuzzy quantizations (or granulations) of variables in fuzzy systems are also examples of fuzzy pseudo partition.

Some of the objective function-based clustering algorithms includes, amongst others, the

- *Fuzzy c-means algorithm (FCM)*: It generates spherical clusters of approximately the same size.
- *Gustafson-Kessel algorithm (GK)*: It provides ellipsoidal clusters with approximately the same size. There are also axis-parallel variants of this algorithm. The algorithm can also be used to detect lines (to some extent).
- *Gath-Geva algorithm (GG)/Gaussian mixture decom*position *(GMD)*: This is used to generate ellipsoidal clusters with varying size. This algorithm has also

axis-parallel variants. The algorithm can also be used to detect lines (to some extent).

- *Fuzzy c-varieties algorithm (FCV)*: It is generally used for the detection of linear manifolds (infinite lines in 2D).
- *Adaptive fuzzy c-varieties algorithm (AFC)*: It can be used to detect line segments in 2D data.
- *Fuzzy c-shells algorithm (FCS)*: It is used for detection of circles (no closed form solution for prototypes).
- *Fuzzy c-spherical shells algorithm (FCSS)*: It detects circular patterns in data.
- *Fuzzy c-rings algorithm (FCR)*: It is also used for detection of circles.
- *Fuzzy c-quadric shells algorithm (FCQS)*: It is used for detection of rectangles (and variants thereof).

Now, let us focus on fuzzy C-mean algorithm.

2.8 FUZZY C-MEANS (FCM) ALGORITHM

The most frequently used fuzzy clustering algorithm is the Fuzzy c-Means (FCM) which is a fuzzification of the k-means algorithm. FCM is a method of clustering which allows one piece of data to belong to two or more clusters. This method was originally proposed by Dunn in 1973 and then improved upon by Bezdek in 1981.

As discussed in Section 2.6, the problem of fuzzy clustering is to find a fuzzy pseudo partition and the associated cluster centers by which the structure of the data is represented as best as possible. This requires some criterion expressing the general idea that associations (in the sense described by the criterion) be strong within clusters and weak between clusters. To solve the problem of fuzzy clustering, the criterion can be formulated in term of a *performance index* which is usually based upon cluster centers. Given a pseudo partition $P = \{A_1, A_2, \ldots, A_c\}$, the cluster centers, $v_1, v_2, \ldots, v_c$ associated with the partition are calculated by the formula given below:

$$v_i = \frac{\sum_{k=1}^{n} [A_i(x_k)]^m x_k}{\sum_{k=1}^{n} [A_i(x_k)]^m}$$

for all $i \in N_c$, where $m > 1$ is a real number that governs the influence of membership grades. It can be observed in the above equation that the cluster center v_i of a fuzzy class A_i is actually the weighted average of data in A_i. The weight of a datum x_k is the m^{th} power of the membership grade of x_k in the fuzzy set A_i.

The performance index of a fuzzy pseudo partition P, $J_m(P)$, is then defined in terms of the cluster centers by the following formula.

$$J_m(P) = \sum_{k=1}^{n} \sum_{j=1}^{c} \left[A_i(x_k)^m\right] \|x_k - v_i\|^2$$

where $\|*\|$ is some inner product-induced norm in space R^p and $\|x_k - v_i\|$ represents the distance between x_k and v_i. This performance index measures the weighted sum of distances between cluster centers and elements in the corresponding fuzzy clusters. Clearly, the smaller the values of $J_m(P)$, the better the fuzzy pseudo partition P. Therefore, the goal of the fuzzy c-means clustering method is to find a fuzzy pseudo

partition P that minimizes the performance index $J_m(P)$. That is, the clustering problem is an optimization problem.

Fuzzy c-means Algorithm

Input: Desired number of clusters c, a real number $m \in (1, \infty)$ and a small positive number ε, serving as a stopping criterion.

Output: A fuzzy pseudo partition and the associated cluster centers.

Steps:

1. Let $t = 0$. Select an initial fuzzy pseudo partition $P^{(0)}$.

2. Calculate the c cluster centers $v_1^{(t)}, v_2^{(t)}, \ldots, v_c^{(t)}$, by using the equation

$$v_i = \frac{\sum_{k=1}^{n} [A_i(x_k)]^m x_k}{\sum_{k=1}^{n} [A_i(x_k)]^m}$$

   For $p^{(t)}$ and the chosen value of m.

3. Update $p^{(*)}$ by the following procedure: For each $x_k \in X$, if $\left\| x_k - v_i^{(t)} \right\|^2 > 0$ for all $i \in N_c$, then define

$$A_i^{(t+1)}(x_k) = \left[\sum_{j=1}^{c} \left(\frac{\left\| x_k - v_i^{(t)} \right\|^2}{\left\| x_k - v_j^{(t)} \right\|^2} \right)^{\frac{1}{m-1}} \right]^{-1}$$

   If $\left\| x_k - v_i^{(t)} \right\|^2 = 0$ for some $i \in I \subseteq N_c$, then define $A_i^{(t+1)}(x_k)$ for $i \in I$ by any nonnegative real number satisfying

$$\sum_{i \in I} A_i^{(t+1)}(x_k) = 1$$

   And define $A_i^{(t+1)}(x_k) = 0$ for $i \in N_c - I$

4. Compare $P^{(t)}$ and $P^{(t+1)}$. That is, compute the distance $| P^{(t+1)} - P^{(t)} |$ between $P^{(t)}$ and $P^{(t+1)}$ in the space $R^{n \times c}$. If $| P^{(t+1)} - P^{(t)} | \leq \varepsilon$, then stop; otherwise, increase t by one and return to step 2.

In the fuzzy c-means algorithm, the parameter m is selected according to the problem under consideration. When $m \to 1$, the fuzzy c-means converges to a "generalized" classical c means. When $m \to \infty$, all clusters tend towards the centroid of the data set X i.e., is the partition becomes fuzzier with increasing m. Currently, there is no theoretical basis for an optimal choice for the value of m. However, it is established that the algorithm converges for any $m \in (1, \infty)$.

We will now illustrate the functioning of the algorithm using a simple one-dimensional data set.

Example 2: Let us suppose the data points are distributed on an axis, say x-axis as shown in Fig. 4.

Fig. 4: Data points and their positions.

Visual inspection of Fig. 4 reveals the existence of two prominent clusters in the data which may be centered around the two locations where the points are more concentrated than other places. Suppose these two clusters are named A and B, and their divisions are shown with the dotted line in Fig. 5. Using the partitional hard clustering algorithm, i.e. the k-means algorithm, each data point will be associated to either cluster A or B. Hence if a membership function is drawn to represent the membership of a datum to cluster A, the function would look as shown in Fig. 5. It shows that points belonging to cluster A have membership value 0 to cluster B.

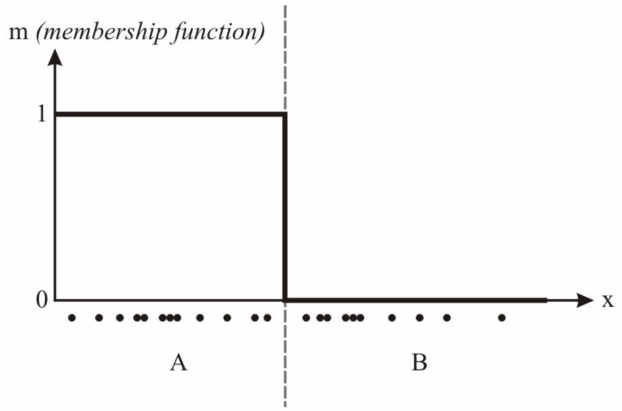

Fig. 5: Membership function for partitional hard clustering of data.

With the Fuzzy c means clustering algorithm, however each datum may not belong exclusively to one cluster. Rather a datum may be placed in both the clusters. A possible membership function for cluster A that can model this situation is shown below. The datum shown as a red spot belongs more to cluster B than to cluster A. Its membership value to cluster A is 0.2.

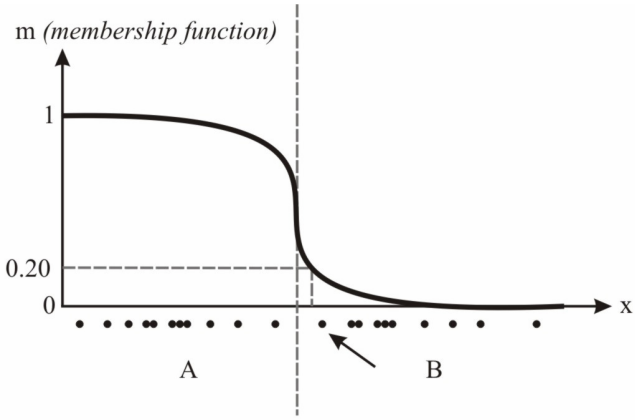

Fig. 6: Fuzzy membership function for cluster 'A'.

Now, instead of using a graphical representation, a matrix $U = [A_i\ (x_k)]$ can be used to represent the membership values to the different clusters. Matrix U will have as many rows as the number of data points and as many columns as the number of clusters chosen. For a hard clustering approach, the values in this matrix are either 0 or 1, as shown by the left hand side matrix for $c = 2$. For a fuzzy clustering matrix however, these values lie between 0 and 1. For example, the right side of Fig. 7 shows that the first datum has membership values 0.8 and 0.2 to clusters A and B, respectively, while the second datum had membership values 0.3 and 0.7 to clusters A and B, respectively.

$$U_{n\times c} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ \cdots & \cdots \\ 0 & 1 \end{bmatrix} \quad U_{n\times c} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ \cdots & \cdots \\ 0.1 & 0.1 \end{bmatrix}$$

Fig. 7: Data points and their positions.

Following is another example, where a set of one-dimensional data points are divided into three clusters. Starting with an arbitrary membership value as shown in Fig. 8(a), the Fig. 8(b) and 8(c) show how the fuzzy membership values evolve, when the algorithm is implemented with a value of 2 for m. The color of the data is that of the nearest cluster according to be membership function. The final convergence shown in Fig. 8(c) had occurred for $\max_{ik}\left\{\left| A_i(x_k)^{(t+1)} - A_i(x_k)^{(t)} \right|\right\} < 0.3$ which was achieved after 37 iterations. It should be kept in mind that different initializations may lead to different final membership values. It may also be the case that starting with different initializations, the same membership values may be obtained, though with a different number of iterations.

The value of m controls the smoothness of the membership function curves. For a large value of m, the transition values of membership from one cluster to another cluster are smoother than those obtained with lower values of m. When $m = 1$, the transition is abrupt and represents the hard clustering technique. Fig.9(a) and (b) show two membership curve for $m = 1.5$ and $m = 2$, respectively.

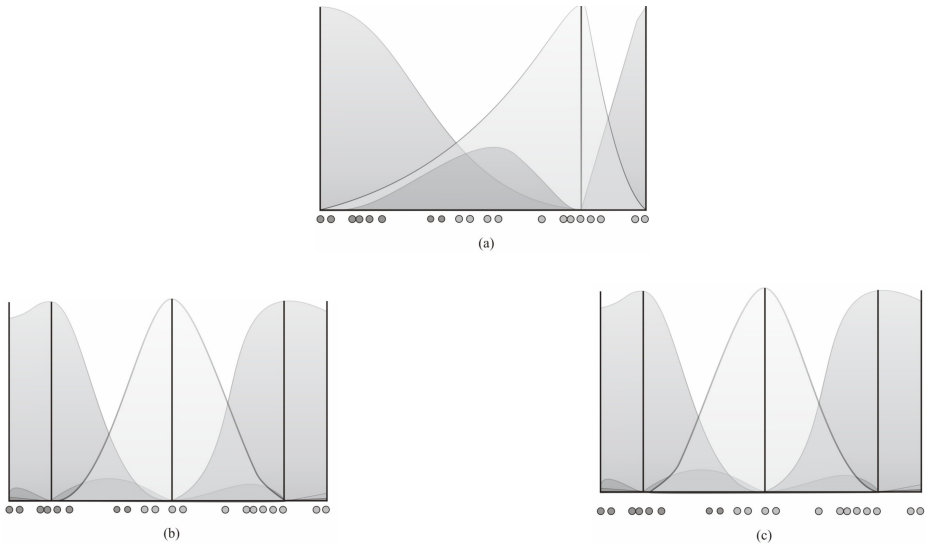

Fig. 8: (a) Initial cluster membership functions for 3 clusters; (b) Intermediate cluster membership functions for 3 clusters; and (c) Final cluster membership functions for 3 clusters.

Fig. 9: Fuzzy clustering membership functions for (a) m = 1.5; and (b) m = 2.

Try the following exercises.

E1) Consider a dataset of six points given in the following table, each of which has two features f_1 and f_2. Assuming the values of the parameters c and m as 2 and the initial cluster centers $v_1 = (5, 5)$ and $v_2(10, 10)$, apply FCM algorithm to find the new cluster center after one iteration.

	f_1	f_2
x_1	2	12
x_2	4	9
x_3	7	13
x_4	11	5
x_5	12	7
x_6	14	4

E2) Consider the following two-dimensional dataset that consists of 15 points in R^2.

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_{k1}	0	0	0	1	1	1	2	3	4	5	5	5	6	6	6
x_{k2}	0	2	4	1	2	3	2	2	2	1	2	3	0	2	4

Assume that we want to determine a fuzzy pseudo partition with two clusters (i.e., $c = 2$). Assume further that $m = 1.25$; $\|*\|$ is the Euclidean distance, and the initial fuzzy pseudo partition is $P^{(0)} = \{A_1, A_2\}$ with

$$A_1 = .854/x_1 + .854/x_2 + \cdots + .854/x_{15}$$

$$A_2 = .146/x_1 + .146/x_2 + \cdots + .146/x_{15}$$

Starting with the initial membership values given above, obtain the final fuzzy pseudo partition and the cluster centers assuming that convergence is achieved when the difference between the two values is ≤ 0.01.

Let us now summarize the unit.

2.9 SUMMARY

In this unit, we have discussed the fundamental concepts of the fuzzy clustering and their applications. We have also discussed how fuzzy clustering differ with classical clustering process. Specifically we have covered the following:

1. Elaborated the essence of clustering and that of fuzzy clustering to solve real life complex problems.
2. Provided a brief introduction of the distance measures used to measure the closeness of the objects.
3. Provided a comparative introduction of the classical clustering and fuzzy clustering.
4. Introduce different classes of clustering algorithms.
5. Provided details about fuzzy clustering algorithms with focus on fuzzy c-means algorithms.

2.10 SOLUTIONS/ANSWERS

E1) The initial membership values of the data point, x_1, in the two clusters can be calculated by using the equation:

$$\mu_{c1}(x_1) = \frac{1}{\sum_{j=1}^{2} \left(\frac{\| x_1 - v_1 \|^2}{\| x_1 - v_1 \|} \right)^2}$$

$$\| x_1 - v_1 \|^2 = 3^2 + 7^2 = 9 + 49 = 58$$

$$\| x_1 - v_2 \|^2 = 8^2 + 2^2 = 64 + 4 = 68$$

$$\mu_{c1}(x_1) = \frac{1}{\frac{58}{58} + \frac{58}{68}} = \frac{1}{1 + 0.853} = 0.5397$$

Similarly, we obtain the following:

$$\mu_{c2}(x_1) = \frac{1}{\frac{68}{58} + \frac{68}{68}} = 0.4603$$

$$\mu_{c1}(x_2) = \frac{1}{\frac{17}{17} + \frac{17}{37}} = 0.6852$$

$$\mu_{c2}(x_2) = \frac{1}{\frac{37}{17} + \frac{37}{37}} = 0.3148$$

$$\mu_{c1}(x_3) = \frac{1}{\frac{68}{68} + \frac{68}{18}} = 0.2093$$

$$\mu_{c2}(x_3) = \frac{1}{\frac{18}{68} + \frac{18}{18}} = 0.7907$$

$$\mu_{c1}(x_4) = \frac{1}{\frac{36}{36} + \frac{36}{26}} = 0.4194$$

$$\mu_{c2}(x_4) = \frac{1}{\frac{26}{36} + \frac{26}{26}} = 0.5806$$

$$\mu_{c1}(x_5) = \frac{1}{\frac{53}{53} + \frac{53}{13}} = 0.197$$

$$\mu_{c2}(x_5) = \frac{1}{\frac{13}{53} + \frac{13}{13}} = 0.803$$

$$\mu_{c1}(x_6) = \frac{1}{\frac{82}{82} + \frac{82}{52}} = 0.3881$$

$$\mu_{c2}(x_6) = \frac{1}{\frac{52}{82} + \frac{52}{52}} = 0.6119$$

Now by using equation $v_1 = \dfrac{\sum_{k=1}^{6} (\mu_{c1}(x_k))^2 \times x_k}{\sum_{k=1}^{6} (\mu_{c1}(x_k))^2}$ the new co-ordinate values for the centre v_1 can be calculated as:

$$\frac{0.5397^2 \times (2,12) + 0.6852^2 \times (4,9) + 0.2093^2 \times (7,13) + 0.4194^2 \times (11,5) + 0.197^2 \times (12,7) + 0.3881^2 \times (14,4)}{0.5397^2 + 0.6852^2 + 0.2093^2 + 0.4194^2 + 0.197^2 + 0.3881^2}$$

$$= \frac{732761}{1.0979}, \frac{10.044}{1.0979} = (6.6273, 9.1484)$$

Similarly, by using the equation $v_2 = \dfrac{\sum_{k=1}^{6} (\mu_{c2}(x_k))^2 \times x_k}{\sum_{k=1}^{6} (\mu_{c2}(x_k))^2}$ the new co-ordinate values for the center v_2 can be calculated as:

$$\frac{0.4603^2 \times (2,12) + 0.3148^2 \times (4,9) + 0.7909^2 \times (7,13) + 0.5806^2 \times (11,5) + 0.803^2 \times (12,7) + 0.6119^2 \times (14,4)}{0.4603^2 + 0.3148^2 + 0.7909^2 + 0.5806^2 + 0.803^2 + 0.6119^2}$$

$$= \frac{22.326}{2.2928}, \frac{19.4629}{2.2928} = (9.7374, 8.4887)$$

E2) The algorithm stops for $t = 6$, and we obtain the fuzzy pseudo partition defined in the following table:

Fuzzy pseudo partition $P^{(6)} = \{A_1, A_2\}$

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$A_1(x_k)$	0.99	1	0.99	1	1	1	0.99	0.47	0.01	0	0	0	0.01	0	0.01
$A_2(x_k)$	0.01	0	0.01	0	0	0	0.01	0.53	0.99	1	1	1	0.99	1	0.99

The two cluster centers are: $v_1 = (0.88, 2)$ and $v_2 = (5.14, 2)$.

2.11 PRACTICAL ASSIGNMENT

Session 1 & 2

Implement the FCM algorithm, using C/C++ language and test it to find the final fuzzy pseudo partition and cluster centers for the two-dimensional data sets given in Table 1 and 2, assuming that convergence is achieved when the difference between the two values is less than or equal to 0.05. Plot the data to determine how many clusters you should assume. Starting with different initial membership matrices check whether the final membership functions are always same? After how many iterations did you achieve the final value in each case? What can you say about the clusters?

Table 1 (Dataset-1)

k	1	2	3	4	5	6	7	8	9	10
x_{k1}	0.958	1.043	1.907	0.780	0.579	0.003	0.001	0.014	0.007	0.004
x_{k2}	0.003	0.001	0.003	0.002	0.001	0.105	1.748	1.839	1.021	0.214

Table 2 (Dataset-2)

k	k	1	2	3	4	5	6	7	8
x_{k1}	x_{k1}	0.958	1.043	1.907	0.780	0.579	0.003	0.001	0.014
x_{k2}	x_{k2}	0.003	0.001	0.003	0.352	0.556	0.105	1.748	1.839

UNIT 3 FUZZY PATTERN RECOGNITION

Structure **Page No**

3.1 INTRODUCTION

Pattern Recognition (PR) form a major area of research and development that encompasses the processing of pictorial, numeric and non-numeric data obtained from the nature. A pattern can be either a physical object (e.g., a human being or a car) or an abstract notion (e.g., a style of writing).

A pattern is represented as a measurement or attribute vector, where each dimension corresponds to a feature of the pattern. These features can be either quantitative or qualitative. For example, if *mass* and *color* are the two features used, then object with 20 units of mass and black in color, can be represented as an ordered two-tuple (20, black). One of the most difficult and challenging problems in pattern recognition is to decide on the ideal set of features for representing a domain. There is no guideline which suggests the appropriate features to use in representing patterns. A major problem is that representation is a highly subjective activity and so a general theory is unlikely to emerge. For example, given a collection of human beings represented using the *mass* attribute alone, it is not possible to separate these patterns into groups corresponding to *tall* and *short* people.

Pattern Recognition techniques can be classified into two broad categories: *unsupervised* techniques and *supervised* techniques. In supervised techniques, a set of human-annotated training samples is used to train the pattern recognition system. During the process of training, the system learns to distinguish between the different classes of samples by associating a distinctive set of feature values to each class. The resulting knowledge is stored as a classifier which can be a set of rules, a mathematical function or a set of weights, depending on the type of system implemented. The accuracy of the classifier is computed by testing it on a set of unseen patterns with known classes. We shall discuss unsupervised pattern recognition and supervised pattern recognition in Section 3.2 and 3.3, respectively.

In traditional pattern recognition systems, an object is classified into one known class. The main disadvantage of traditional (crisp) set theory is that it implies an aura of precision and definiteness for a decision that may not be warranted. In real-world systems, it is difficult to arrive at a crisp distinction always. Often data in a "boundary" condition can contain samples that could be said to be a member of more than one class. It can very well happen that a boundary condition sample X may be assigned to class A using crisp technique T1, but to class B using another crisp

technique T2, and there is no way to gauge from the results which assignment is more appropriate.

Fuzzy pattern recognition systems on the other hand assign fuzzy membership values to all classes for all objects. Let us consider an example where existing bank customers have to be potentially classified into two groups – loan-worthy or non loan-worthy customers, based on their past transaction patterns. While there may be a well-defined set of features that can decide whether a customer is loan-worthy or not, it is not likely that a customer will necessarily satisfy all the properties of any one or the other class. In such a scenario, it is more practical to go for a fuzzy classification and exploit the fuzzy membership values to the two classes as some kind of confidence measure.

An unsupervised technique, on the other hand, does not use a given set of pre-classified data points. Since there is no pre-defined set of classes to associate patterns, this process is usually applied for grouping similar patterns together. Clustering is one of the primary techniques for grouping similar patterns together. We have already discussed Fuzzy clustering in Unit 2. In this unit, we will discuss some applications of clustering to unsupervised pattern recognition. Supervised pattern recognition and knowledge-based pattern recognition using fuzzy logic will follow in Section 3.4. Finally, we describe hybrid approaches and an application of hybrid fuzzy systems in medical image segmentation in Section 3.5. At the end we shall discuss multivalued recognition system in Section 3.6.

Objectives

After studying this unit, you should be able to:

- know the essence of the concept of pattern recognition;
- know what different types of PR systems are;
- differentiate between the supervised and unsupervised pattern recognition systems;
- know the essence of the hybrid system to solve complex PR problems;
- apply fuzzy logic to solve real-life PR problems.

3.2 UNSUPERVISED PATTERN RECOGNITION

Unsupervised pattern recognition is motivated by the need to find interesting patterns or groupings in a given set of data, the class labels of which are not known. For example, the manager of a supermarket may be interested in collecting data about a group of customers and analyzing it to find interesting grouping of customers on the basis of their buying patterns. The result of such an analysis can be used by a company to develop its marketing strategy or to develop a family of product lines that are targeted toward various customer groupings.

Image segmentation refers to partitioning pixels on an image into regions that correspond to different objects or different faces of objects in the images. Image segmentation can be viewed as data clustering problem where each datum is described by a set of image features like intensity, color, texture, etc. of a pixel. Conventional clustering algorithm generally finds a “hard partition” of a given dataset but in many real-world clustering problems, some data points partially belong to multiple clusters, rather than to a single cluster exclusively. For example, a pixel in a Magnetic Resonance Image (MRI) may correspond to a mixture of two different types of tissues. Another area which uses a lot of fuzzy clustering is that of customer behavior analysis. Clustering aims at finding similar groups of customers to target a particular product or promotional campaign. However, a particular customer may be a “borderline case”

between two groups of customers (e.g., between moderate conservatives and moderate liberals). Fuzzy clustering provides a method of handling such borderline or ambiguous cases with ease. "Fuzzy clustering" algorithms attempt to find a "soft partition" of a given dataset in which a datum can partially belong to multiple clusters. One of the most popular fuzzy clustering algorithms, Fuzzy c-Means algorithm, has been already discussed in Unit-2. In the following sub-section an application of this algorithm is presented.

Applications of Fuzzy Clustering to Image Segmentation

One of the oldest applications of fuzzy clustering techniques has been to Image Segmentation. Algorithms such as fuzzy c-means (Bezdek, 1981) have been used to build clusters (segments). The class membership of pixels can be interpreted as similarity or compatibility with an ideal object or a certain property. Prior to applying fuzzy clustering techniques for image segmentation, it is necessary to perform image processing using fuzzy logic. This occurs in three stages (http://pami.uwaterloo.ca/tizhoosh/segment.htm). The first step is known as *image fuzzification,* and this is used to modify the membership values of a specific data set or the image. This step results in the transformation of the image data from gray-level plane to the membership plane. In the second step, appropriate fuzzy techniques are applied to modify the membership values. These techniques could be based on fuzzy clustering, a fuzzy rule-based approach, or a fuzzy integration approach. The final stage comprises of decoding of the results, called *defuzzification*. This step results in an output image. The main power of fuzzy image processing lies in modification of the fuzzy membership values. Fig. 1 below shows the steps involved. The roles of these steps are further explained through the following example.

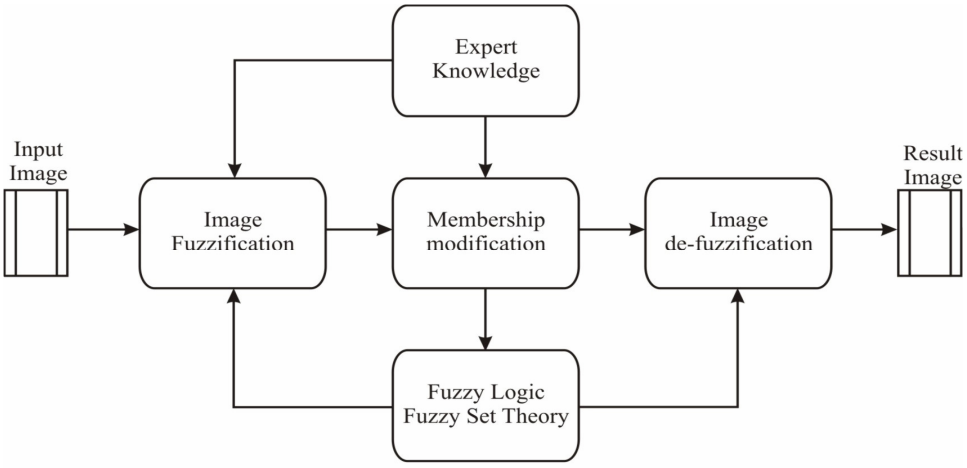

Fig. 1: Steps in Fuzzy Image Processing.

Example 1: Let us suppose an image comprises of grayscale values ranging from 0 to 255. It is required to segment the image into two regions – one consisting of the darker regions in the image and the other containing the comparatively lighter one. The image has different levels of darkness, and in classical set theory, a threshold such as the gray level 100 has to be set, so that if a pixel's gray-level value is less than 100 it will fall in the dark region, otherwise in the lighter region. But since the darkness of a particular pixel is a matter of degree, a fuzzy set (or subset to be precise) can model this property much better. To define this subset, two thresholds, say gray levels 50 and 150 are required. Then all the gray levels that are less than 50 are full member of the set dark, and all gray levels greater than 150 are definitely not members of the set. Gray levels between 50 and 150, however, have a partial membership to the set dark.

In fig. 2, a test image containing a few machine parts are shown. It is required to segment these parts out from the background. It is obvious that since the parts have different levels of grayness, it is difficult to choose a threshold value by inspection,

which can separate the black background from the foreground objects. Fuzzy image processing can be used here to do the job of separating the background from the foreground objects successfully.

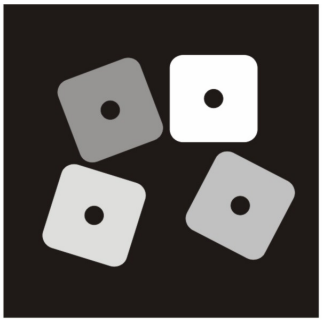

Fig. 2: Test Image for segmenting into background and foreground objects.

Fig. 3 below shows how fuzzy membership using standard S function can be used to decide the threshold for segmentation. The membership function, S function in this case, is moved pixel by pixel over the existing range of gray levels. In each position, a measure of fuzziness is calculated. The S function operates on each pixel with co-ordinate (m,n), and gray-scale value μ_{mn} as follows:

$$\mathbf{S}(\mu_{mn}) = -\mu_{mn}\ \ln(\mu_{mn}) - (1-\mu_{mn}).$$

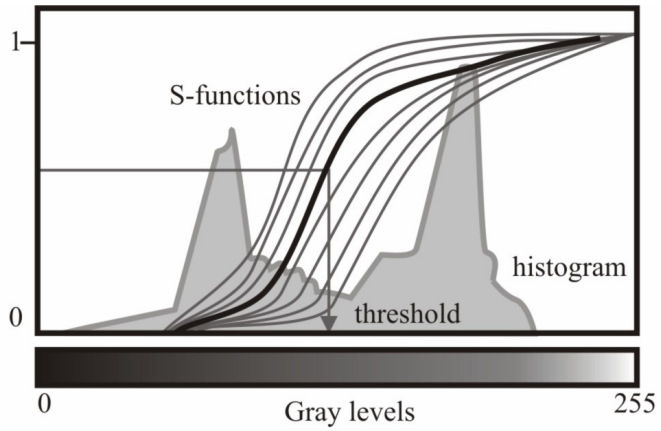

Fig. 3: Choosing threshold using S function for Image fuzzification.

The position with a minimum amount of fuzziness, as shown in the figure, is then chosen as the threshold. All pixels with fuzzy value greater than this threshold is assigned the value 255, while all pixels with fuzzy value less than this is assigned the value 0. The resulting image is shown in fig. 4. The image is clearly segmented into the background and the foreground objects.

The resulting image is shown in fig. 4. The image is clearly segmented into the background and the foreground objects.

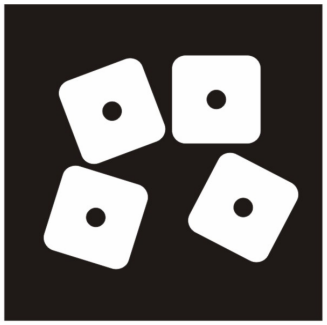

Fig. 4: Segmented image obtained using fuzzy techniques.

Now, let us focus on supervised pattern recognition.

3.3 SUPERVISED PATTERN RECOGNITION

In the previous section, we have discussed unsupervised pattern recognition technique which does not require points in the dataset to be labeled with their correct classification. In contrast, supervised pattern recognition uses data with known classifications, which are also called labeled data, to determine the classification of new data.

Supervised and unsupervised pattern recognition techniques differ in their advantages and limitations. The main benefit of unsupervised pattern recognition technique is that they do not require training data; however, their computation time is relatively high due to a large number of iterations needed before the algorithm converges. In contrast, a supervised pattern recognition technique is relatively fast because it does not need to iterate; however, it cannot be applied to a problem unless training data or relevant knowledge are available.

Now, let us discuss fuzzy k-nearest neighborhood.

Nearest neighbour (NN) classification techniques classify an unknown sample by comparing it to its nearest neighbors among a set of known samples. Any distance metric can be used for computing the nearest neighbours of an object, as long as it applies consistently to all samples in the set. An unknown sample is assigned membership to the class most represented by its K nearest neighbours.

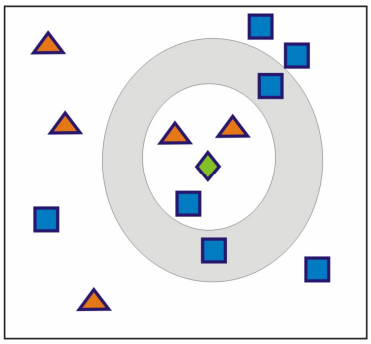

Fig. 5: Classification problem–Two classes of objects are known: triangles and squares. Find the class of the unknown object diamond shown in the centre.

Fig. 5 shows how a classification task is posed and solved using K–NN. Let us suppose, the system is trained to recognize two classes of objects–triangles and squares. In order to determine the class of an unknown object, the diamond, using K– NN and setting K to 3, it can choose three of its closest neighbours as shown by the inner circle. In this case, the class assigned to the diamond would be triangle. However, if K is chosen to be 5, then the diamond will be classified as a square.

Crisp K-NN classification techniques can be computationally quite simple and can run in O(N) time. However, crisp K–NN classification techniques suffer from major drawbacks as elaborated below:

a) An object can be classified into only one class based on the class of majority of its neighbours. Noise in the data can really lead to a very erroneous classification in this case, since all neighbours are given equal weights.

b) In case of a tie among more than one class, decision has to be arbitrarily taken in favour of one class or the other without any rationale.

For fuzzy K-NN, an unknown sample's membership to each class is assigned based on the neighbour's class and the distance of the neighbour from the unknown sample.

In essence, membership of object ***x*** to class ***i*** is given by the following equation, where K is the total number of neighbours considered.

$$\mu_i(x) = \frac{\sum_{j=1}^{k} \mu_{ij} \left(\frac{1}{\| x - x_j \|^{\frac{2}{m-1}}} \right)}{\sum_{j=1}^{k} \left(\frac{1}{\| x - x_j \|^{\frac{2}{m-1}}} \right)}$$

The denominator in the above equation acts as a normalizing factor. This factor decides how the distance of a neighbour from the sample should influence the classification decision. When m is chosen to be greater than 1, the farther a neighbour is from an object, the less is its influence on the classification decision. For m greater than 1, this effect reduces exponentially. Though there is no theoretical study on the appropriate value of m to be chosen, m is usually chosen to be 2.

In the above example, the squares and the triangles would contribute almost equally to the class of the unknown sample diamond, and the object would have memberships to both classes, which is more sensible in the given scenario.

In the following section, we shall discuss knowledge-based pattern recognition.

3.4 KNOWLEDGE-BASED PATTERN RECOGNITION

A knowledge-base comprises an information store and an inference engine for making decisions. Fuzzy pattern recognition techniques have been very successful in implementing Knowledge-based systems for analyzing sensor data, where the nature of the data is such that non-fuzzy techniques cannot isolate patterns effectively. Sensor data is usually noisy. In this section, we shall discuss an application where fuzzy techniques have been successfully employed for real-time decision making in an automated manufacturing unit (Singh and Steinl, 1996).

Automation of factories is principally related to the automation of individual processes and decision making related to their operation. Decision making systems are typically designed as knowledge-based systems, where making decisions is based on a set of conditions and rules. The conditions are affected by the sensor data. These systems can become very complicated as their size increases. The increase in size is directly proportional to the number of components, the number of sensor measurements per component and the number of sensors. Large knowledge-bases are expensive to manage and modify. Decision-making in a large set-up is also complex, and the task of real-time decision making is challenging since decisions need to be made in such a manner that there are no bottlenecks on the production-line. Fuzzy search technique is related to the concept of possibility measurements in fuzzy logic, which can handle incomplete and imprecise data and perform fast searches over a large database.

In order to help real-time decision making, the first step is to reduce the amount of information needed for decision making. Since sensor data is noisy, the first step is to reduce the dataset into bandwidths rather than deal with a large number of unique sensor outputs. For example, if A and B are two sensors, whose outputs can be any real value in the range of 1 to 10 with resolution 0.1, then rather than have rules like *If* $(A = 1 \;\&\; B = 2)$ *OR* $A = 1.1 \;\&\; B = 2)$. *Then Component* 1, a rule which states *If* $(.9 < A < 1.1 \;\&\; 1.9 < B < 2.1)$. *Then Component* 1, is obviously more compact and

useful. This is a simplistic representation which can be further improved as using the concept of possibility distribution as explained below.

The basic concept of a possibility distribution derives from the fact that the mean of a sample has the maximum possibility of occurrence, and this possibility decreases quadratically as one moves away from the mean. For example, let us suppose that for a total of 100 men, their heights are stored in a set H. Let H_{min} represent the minimum height, H_{max} represent the maximum height and H_{mean} the mean height of this set. If it was required to guess a new person's height, based on experience about men's height from the set H, then it is possible to make an assessment in a way such that the assessed height has a membership value which lies between 0 and 1, by making use of possibility distribution function. The possibility distribution function looks similar to a normal curve. Fig. 6 below illustrates the possibility distribution function along with membership function formulae.

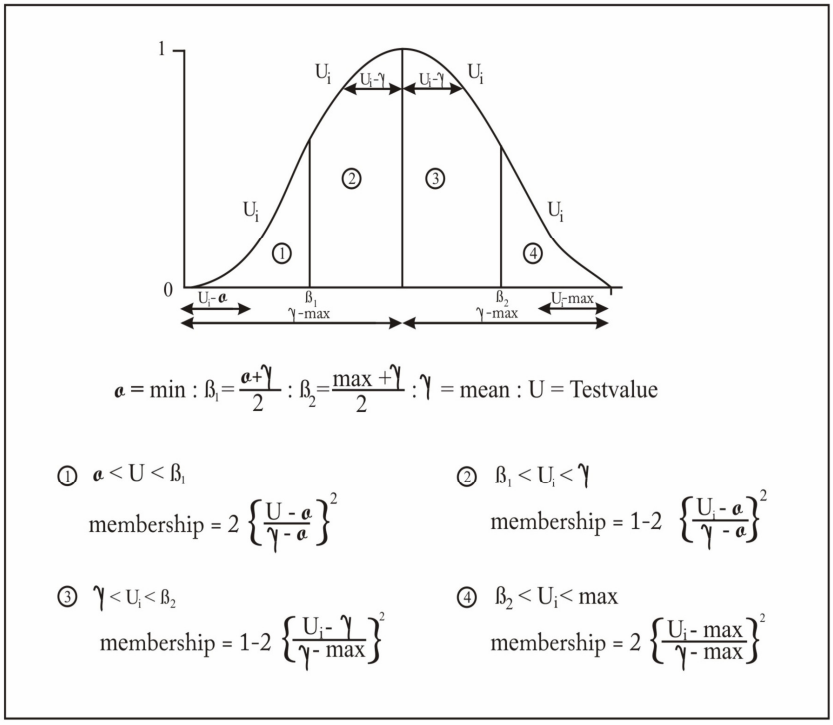

Fig. 6: Possibility Distribution Function for calculating memberships for new values.

The possibility distribution function can be computed to interpret sensor data in real-time, where each sensor has a minimum, mean and a maximum that is stored in the knowledge base. U_i represents a new sensor data, which has to be interpreted by the inference engine.

Let us suppose a manufacturing unit has **N** components (C_1, C_2,…,C_N). Let each component record $\mathbf{k_{max}}$ readings for each sensor to account for sensor reliability, uncertainty, noise, etc. Thus sensor reading for component C_1 has (C_{11}, C_{12}, C_{1kmax}). Each reading consists of past $\mathbf{i_{max}}$ sensor measurements. A new reading $X = (X_1, X_2, \ldots, X_{imax})$ has to be interpreted by the system for taking a decision.

Initially, the system computes for column **i** of component **j**, the minimum, maximum and mean of its k_{max} readings and stores them as min_{ij} , max_{ij} and $mean_{ij}$, respectively. Since this is all that it needs for computing the possibility of a new reading, the actual size of the original data set has been reduced by a factor of $k_{max}/3$. For the individual measurements of X, the possibility of occurrence of each value is computed with respect to each component, is computed as a membership value between (0,1). The membership of new reading X_i with respect to the sensor **i** of component **j** is computed as $M(X_i, C_{ij})$, using the possibility distribution function.

In order to determine the component that readings of vector X represent, the overall possibility for each row has to be computed. This is achieved by calculating a cumulative index of possibility for each component by combining the individual values using the following function:

$$I_j = e^{M(x_1, C_{1j})} + \cdots + e^{M(x_1, C_{1j})} + \cdots + e^{M(x_{i_{max}}, C_{i_{max}})}$$

Index I_j is computed for each row in the knowledge base and the component corresponding to the highest index is assumed to represent X.

Now, let us discuss hybrid pattern recognition systems.

3.5 HYBRID PATTERN RECOGNITION SYSTEMS

The field of pattern recognition has seen enormous progress since its beginning more than 5 decades ago. Over the years various approaches have been emerged, based on statistical decision theory, structural matching and parsing, neural networks, fuzzy logic, artificial intelligence, evolutionary computing and others. Obviously, these approaches are characterized by a high degree of diversity. In order to combine their strengths and avoid their weaknesses, hybrid pattern recognition systems that combine several techniques together have come into existence. The major applications areas of hybrid pattern recognition system include on-line and off-line handwriting recognition, remotely sensed image interpretation, fingerprint identification, and automatic text categorization. In the following sub-section we will present neuro-fuzzy systems that have been successfully applied in medical domain. A neuro-fuzzy system is a fuzzy system that uses a learning algorithm derived from or inspired by neural network theory to determine its parameters (fuzzy sets and fuzzy rules) by processing data samples.

3.5.1 Combining Fuzzy Systems with Neural Networks

Both neural networks and fuzzy systems have some things in common. They can be used for solving a problem (e.g. pattern recognition, regression or density estimation) if there does not exist any mathematical model of the given problem. They solely do have certain disadvantages and advantages which almost completely disappear by combining both concepts.

Neural networks can only come into play if the problem is expressed by a sufficient amount of observed examples. These observations are used to train the *black box*. On the one hand no prior knowledge about the problem needs to be given. On the other hand, however, it is not straightforward to extract comprehensible rules from the neural network's structure.

On the contrary, a fuzzy system demands linguistic rules instead of learning examples as prior knowledge. Furthermore the input and output variables have to be described linguistically. If the knowledge is incomplete, wrong or contradictory, then the fuzzy system must be tuned. Since there is not any formal approach for it, the tuning is performed in a heuristic way. This is usually very time consuming and error-prone.

Table 1 highlights the relative similarities and dissimilarities of the two approaches. Neural networks can learn from data, but cannot be interpreted - they are black boxes to the user. Fuzzy Systems consist of interpretable linguistic rules, but they cannot learn. It is desirable for fuzzy systems to have an automatic adaption procedure which is comparable to neural networks. We use learning algorithms from the domain of neural networks to create fuzzy systems from data. The learning algorithms can learn both fuzzy sets, and fuzzy rules, and can also use prior knowledge.

Table 1: Comparison of Neural and Fuzzy systems

Neural Networks	Fuzzy Systems
no mathematical model necessary	no mathematical model necessary
learning from scratch	a-priori knowledge essential
several learning algorithms	not capable to learn
black-box behaviour	simple interpretation and implementation

3.5.2 Neuro-Fuzzy System or Fuzzy Neural Network

In the field of artificial intelligence, *neuro-fuzzy* refers to combinations of artificial neural networks and fuzzy logic. Neuro-fuzzy hybridization results in a hybrid intelligent system that synergizes these two techniques by combining the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks. Neuro-fuzzy hybridization is widely termed as Fuzzy Neural Network (FNN) or Neuro-Fuzzy System (NFS) in the literature. Neuro-fuzzy system (the more popular term is used henceforth) incorporates the human-like reasoning style of fuzzy systems through the use of fuzzy sets and a linguistic model consisting of a set of IF-THEN fuzzy rules. The main strength of neuro-fuzzy systems is that they are universal approximators with the ability to solicit interpretable IF-THEN rules.

The strength of neuro-fuzzy systems involves two contradictory requirements in fuzzy modeling: interpretability versus accuracy. In practice, one of the two properties prevails. The neuro-fuzzy in fuzzy modeling research field is divided into two areas: linguistic fuzzy modeling that is focused on interpretability, mainly the Mamdani model; and precise fuzzy modeling that is focused on accuracy, mainly the Takagi-Sugeno-Kang (TSK) model.

Characteristics of NFS

Compared to a common neural network, connection weights and propagation and activation functions of fuzzy neural networks differ a lot. Although there are many different approaches to model a fuzzy neural network, most of these agree on certain characteristics such as the following:

- A neuro-fuzzy system based on an underlying fuzzy system is trained by means of a data-driven learning method derived from neural network theory. This heuristic only takes into account local information to cause local changes in the fundamental fuzzy system.
- It can be represented as a set of fuzzy rules at any time of the learning process, i.e., before, during and after. Thus the system might be initialized with or without prior knowledge in terms of fuzzy rules.
- The learning procedure is constrained to ensure the semantic properties of the underlying fuzzy system.
- A neuro-fuzzy system approximates a n-dimensional unknown function which is partly represented by training examples. Fuzzy rules can thus be interpreted as vague prototypes of the training data.
- A neuro-fuzzy system is represented as special three-layer feedforward neural network as it is shown in Fig. 7.
    - The first layer corresponds to the input variables.
    - The second layer symbolizes the fuzzy rules.
    - The third layer represents the output variables.
    - The fuzzy sets are converted as (fuzzy) connection weights.

- Some approaches also use five layers where the fuzzy sets are encoded in the units of the second and fourth layer, respectively. However, these models can be transformed into a three-layer architecture.

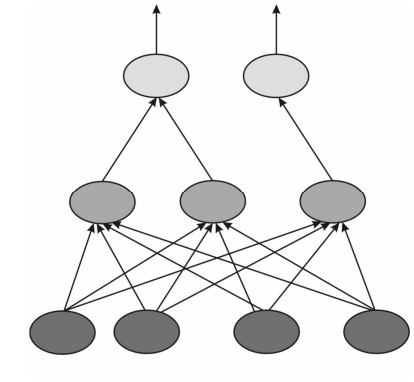

Fig. 7: The layers in a neuro-fuzzy system.

3.5.3 Hybrid Neuro-Fuzzy System or Fuzzy Neural Network

Hybrid neuro-fuzzy systems are homogeneous and usually resemble neural networks. Here, the fuzzy system is interpreted as a special kind of neural network. The advantage of such hybrid NFS is its architecture since both fuzzy system and neural network do not have to communicate any more with each other. They are one fully fused entity. These systems can learn online and offline. Fig. 8 shows such a hybrid FNN.

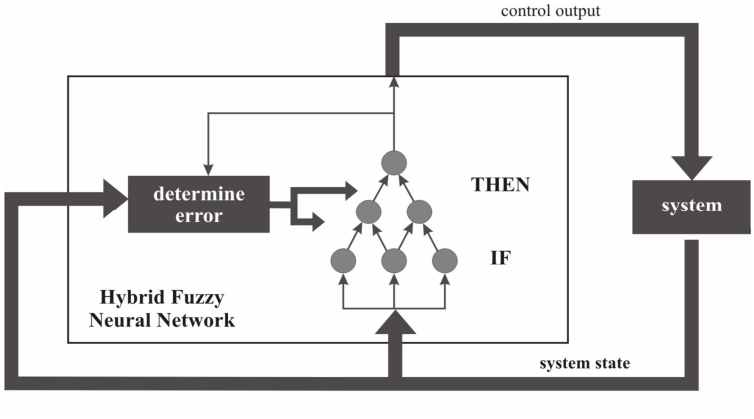

Fig. 8: The architecture of a hybrid fuzzy neural network.

The rule base of a fuzzy system is interpreted as a neural network. Fuzzy sets can be regarded as weights whereas the input and output variables and the rules are modeled as neurons. Neurons can be included or deleted in the learning step. Finally, the neurons of the network represent the fuzzy knowledge base. Obviously, the major drawbacks of both underlying systems are thus overcome.

In order to build a fuzzy controller, membership functions which express the linguistic terms of the inference rules have to be defined. In fuzzy set theory, there does not exist any formal approach to define these functions. Any shape (e.g., triangular, Gaussian) can be considered as membership function with an arbitrary set of parameters. Thus, the optimization of these functions in terms of generalizing the data is very important for fuzzy systems. Neural networks can be used to solve this problem.

By fixing a distinct shape of the membership functions, say triangular, the neural network must optimize their parameters by gradient descent. Thus, apart from the

information about the shape of the membership functions, training data must also be available.

Another approach is to group the training data $\{(X_i, y_i) | x_i \in \chi,\ y_i \in Y,\ i = 1,2,\cdots,l\}$ into M clusters. Every cluster represents a rule R_m where m = 1, 2 ,…, M. Hence these rules are not defined linguistically but rather by crisp data points $X = (x_1, x_2, \ldots, x_n)$ $X = (x_1, x_2, \cdots, x_n)$. Thus, a neural network with n input units, hidden layers and M output units might be applied to train on the pre-defined clusters. For testing, an arbitrary pattern x is presented to the trained neural network. Every output unit m will return a degree to which extent x may fit to the antecedent of rule R_m.

To guarantee the characteristics of a fuzzy system, the learning algorithm must enforce the following mandatory constraints:

- Fuzzy sets must stay normal and convex.
- Fuzzy sets must not exchange their relative positions (they must not *pass* each other).
- Fuzzy sets must always overlap.

Additionally there do exist some optional constraints like the following:

- Fuzzy sets must stay symmetric.
- The membership degrees must sum up to 1.

The ARIC (Approximate Reasoning-based Intelligent Control) is a fuzzy neural network model where a prior defined rule base is tuned by updating the network's prediction. Thus the advantages of fuzzy systems and neural networks are easily combined. ARIC is represented by two feed-forward neural networks, the Action-state Evaluation Network (AEN) and the Action Selection Network (ASN). ASN is a multilayer neural network representation of a fuzzy system. It consists of two separate modules. The first one represents the fuzzy inference and the second one computes a confidence measure based on the current and next system state. Both parts are eventually combined to the ASN's output.

As it is shown in Fig. 8, the first layer represents the rule antecedents, whereas the second layer corresponds to the implemented fuzzy rules and the third layer symbolized the system action. The network flow is as follows. In the first layer the system variables are fuzzified. In the next step these membership values are multiplied by the attached weights of the connections between the first and second layer. In the latter layer, every rule's input corresponds to the minimum of its input connections.

A rule's conclusion is installed as membership function. This function maps the inverse rule input value. Its output values are then multiplied by the weights of the connections between second and third layer. The final output value is eventually computed by the weighted average of all rule's conclusions.

The AEN (which is as three-layer feed-forward neural network as well) aims to forecast the system behaviour. The hidden layer obtains as input both the system state and an error signal from the underlying system. The output of the networks shall represent the prediction of the next reinforcement which depends on the weights and the system state. The weights are changed by a reinforcement procedure which takes into consideration the outputs of both networks ASN and AEN. ARIC was successfully applied to the cart-pole balancing problem.

Whereas the ARIC model can be easily interpreted as a set of fuzzy-if-then rules, the use of ASN network to adjust the weights is rather difficult to understand. It is a

working neural network architecture that utilizes aspects of fuzzy systems. However, a semantic interpretation of some learning steps is not possible.

3.5.4 Applications in Medical Image Segmentation

Image segmentation is defined as the partitioning of an image into non-overlapping regions that are homogeneous with respect to some characteristic such as intensity, which does not easily happen in the medical image because most of medical images have a tissue (overlapping region). Image segmentation has its root in many real-life problems, especially medical applications. The imaging modalities can be divided into two global categories: anatomical and functional. Anatomical modalities, depicting primarily morphology, include X-ray, CT (Computed Tomography), MRI (Magnetic Resonance Imaging), US (ultrasound), portal images, and (video) sequences. Image segmentation algorithms play a role in biomedical imaging applications such as the quantification of tissue volumes diagnosis, localization of pathology study of anatomical structure, treatment planning, partial volume correction of functional imaging data, and computer integrated surgery. The methods for performing segmentations vary widely depending on the specific application. For example, the segmentation of brain tissue has different requirements from the segmentation of the liver. In general, there are two main approaches to clustering–crisp clustering and fuzzy clustering. One of the popular fuzzy clustering techniques proposed in literature for medical image segmentation is fuzzy c-means (FCM) algorithm. The neural network based approaches have been also introduced in literature for medical image segmentation. The use of fuzzy c-means algorithm with Kohonen neural network for MRI medical image segmentation can help to recognize the tumor region and its characteristics. Tsao et al., 1994 have extended the ideas of FCM and Kohonen neural network to a new family of algorithms called Fuzzy Kohonen Clustering Network (FKCN). The Kohonen Clustering Network (KCN) clustering is closely related to the Fuzzy C-Means (FCM) algorithms. Since Fuzzy C-Means algorithms are optimization procedure because the objective function is approximately minimized, the integration of FCM and KCN is one way to address several image segmentation problems. They combine the ideas of fuzzy membership values for learning rates, the parallelism of Fuzzy C-Means, and update rules of KCN. FKCN is a self-organizing algorithm, since the “size” of the updated neighbourhood is automatically adjusted during learning, and FKCN usually terminates in a way of minimized objective function of FCM. Fig. 9 illustrates the application of FKCN to identify brain tumor in a magnetic resonance image of a brain.

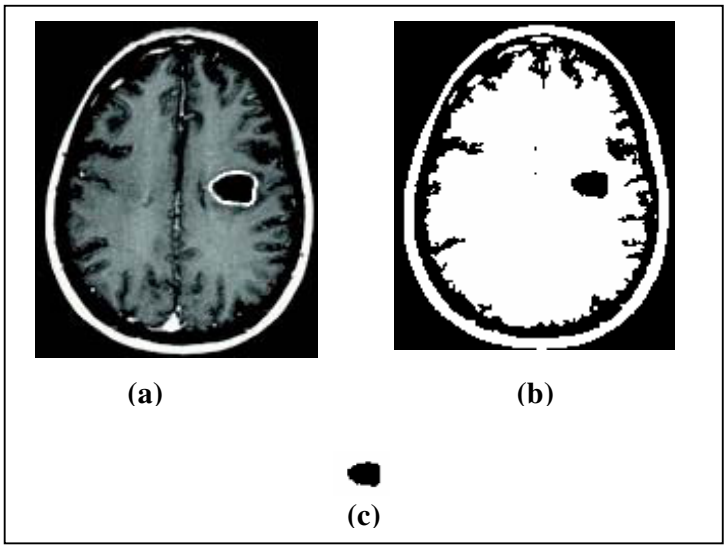

Fig. 9: (a) Magnetic Resonance Image (MRI) of a brain; (b) MRI of the brain after applying Fuzzy Kohonen neural network; and (c) Identified tumor.

Now, let us focus on multivalued recognition systems.

3.6 MULTIVALUED RECOGNITION SYSTEMS

Fuzzy logic is a form of multi-valued logic that is derived from fuzzy set theory. In fuzzy logic the degree of truth of a statement is represented by a value in the range of (0,1) both inclusive. This is in contrast to predicate logic which allows for only two truth values (TRUE, FALSE). Fuzzy logic allows for reasoning that is approximate rather than precise.

To begin with, let us consider designing a system based on anti-lock brakes. The anti-lock brakes are controlled by temperature. The anti-lock brakes have membership functions that control its function depending on different temperature ranges. Fig. 10 below defines how temperature ranges can be used to define membership values to three categories – cold, warm and hot. Fig. 10 states that, low temperature values indicate high membership values for cold and low membership values for hot and warm, medium temperatures indicate high membership value for warm and low membership values for both hot and cold, and high temperature indicates high membership value to hot and low membership values to cold and warm.

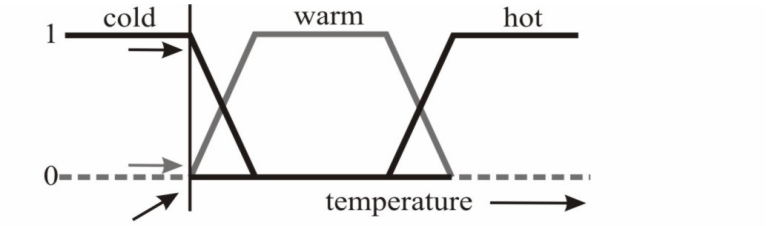

Fig. 10: Fuzzy membership functions to ascertain cold, hot and warm depending on temperature.

Fuzzy logic based systems are usually implemented using IF-THEN rules, though they can be implemented in other ways also. A rule is written in the form of "IF *variable* IS *property* THEN *action*". It may be noted that these rules do not have any ELSE condition.

A simple temperature regulator connected to a fan may be designed using Fuzzy logic using the following rules:

IF *temperature* IS **COLD** then *step down speed of fan*

IF *temperature* IS **WARM** then *maintain normal speed of fan*

IF *temperature* IS **HOT** then *speed up fan*

Given a particular value of temperature, it is possible that more than one antecedent is fired with value greater than 0. For example if the temperature is 15 degrees, both cold and warm have non-zero membership values. Thus, the antecedents of both the first and the second rules have the potential to be fired. The final action will depend on the maximum truth value that the system attains.

The AND, OR and NOT operators are defined in fuzzy logic using *min*, *max* and *complement*. For fuzzy variables *x* and *y*, the truth values of combinations of *x* and *y* are defined as follows:

Truth(*x* AND *y*) = min (truth(*x*), truth(*y*))

Truth(*x* OR *y*) = max (truth(*x*), truth(*y*))

Truth(*NOT*(x)) = 1 – truth(*x*)

Fuzzy logic has been successfully used in edge detection for medical images. Lalande et al. [2] described a system for automatic detection of cardiac contours in MRI images using fuzzy logic. Since the heart is not an isolated organ, traditional edge detection operators do not perform well in such images.

The system uses two parameters. The first parameter ***p*** is estimated from the image using the logic that all pixels located on the cardiac contour will have roughly the same gray-value. To determine the gray level value of the contour pixels, the user indicates a point near the centre of the cardiac left ventricle. Then, a series of radial lines is automatically drawn. Along each line, a Gaussian edge operator (Lalandel et al., 1997) is used to detect the first edge between a high-intensity area and a low-intensity area (i.e. contour between the cardiac cavity and the heart muscle). Along each line the pixel where the Gaussian operator yields the highest value is noted and the gray level value of this pixel is retrieved. The extracted gray level value ***p*** of the endocardial contour is the maximum of the histogram of these local gray level values obtained along all radial lines.

Once the value ***p*** is estimated, all other pixels are compared to this value. The second parameter is computed based on the presence of edges, using fuzzy logic. For a given gray level, the closer the value is to the estimated contour gray value, the greater the likelihood that the component belongs to the myocardium. Fig. 11 describes the membership functions used for this purpose. The edge contour is finally found by linking the edge pixels and detecting the best path. Fig. 12 highlights the results.

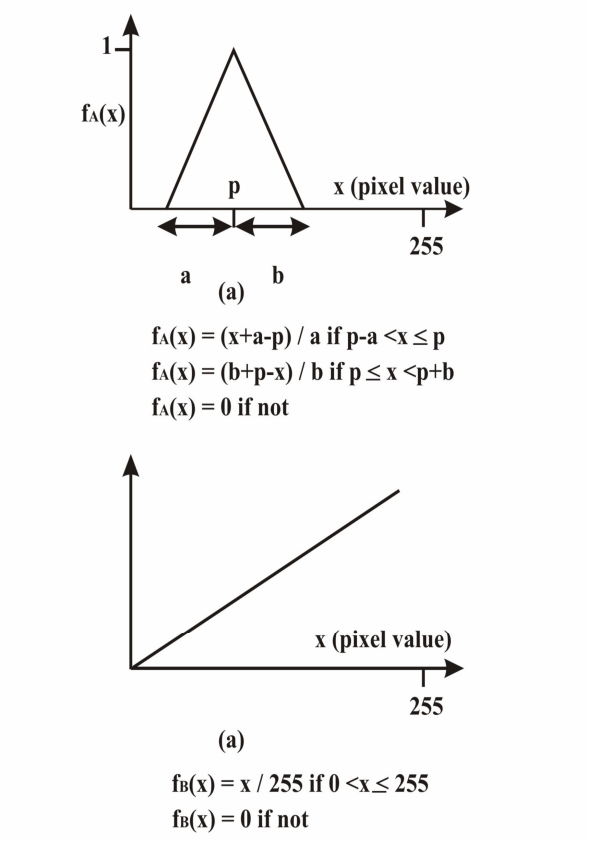

Fig. 11: (a) Membership function to determine whether a pixel belongs to the myocardium (A) ; and (b) Membership function to determine whether a pixel belongs to the cardiac contour.

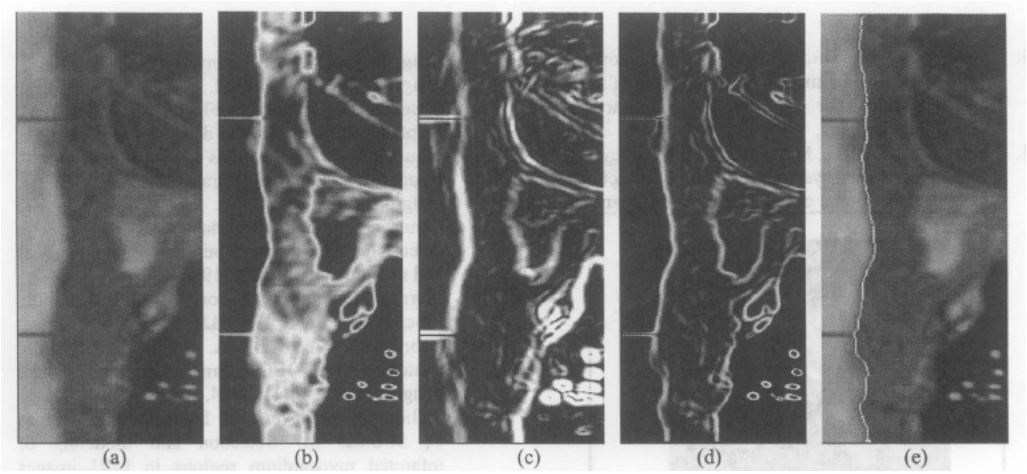

Fig. 12: (a) Initial image; (b) Membership degrees associated with gray level pixels; (c) Membership degrees associated with edges; (d) Edge pixels; and (e) Final edge contour after linking.

Now try the following exercises.

E1) John Doe is a medical doctor interested in statistics. He uses a certain test to determine if his patient Mr. Saha has cancer. However, the test is not perfectly reliable. In fact, the test succeeds to reveal cancer only with probability 0.98. In addition, when the patient does not have cancer, the test has a probability of 0.03 to be positive. It is also known that the probability of cancer in the overall population is 0.008.

What is the probability that Mr. Saha has cancer when the test is positive? After the positive result, John Doe decides to perform the test once more. The second test also results positive. What is the probability of cancer after the second test? Assume that the results of the tests are independent.

E2) Consider two normally distributed probability distributions

$$p\left(x\middle|w_i\right)=\frac{1}{\sqrt{2\pi\sigma}}\exp\left[-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma}\right)^2\right]$$

with equal deviations $\sigma=1$ and a priori probabilities $P\left(\omega_1\right)=P\left(\omega_2\right)$.
Determine a classifier with a minimum classification error.

Now, let us summarize the unit.

3.7 SUMMARY

In this unit, we have discussed the fundamental concepts of the fuzzy pattern recognition and its various real-life especially medical applications. We have also discussed how the supervised pattern recognition differs with unsupervised pattern recognition process. Specifically we have covered the following:

1. Elaborated the essence of pattern recognition and its applications to solve real life complex problems.
2. Explained both supervised and unsupervised processes with examples for pattern recognition.

3. Discussed knowledge-based pattern recognition system that uses domain knowledge to improve the quality of classification process.
4. Discussed how fuzzy system and neural networks can be combined into a hybrid pattern recognition to combine the strengths of both approaches for solving complex pattern recognition problems.
5. Provided details about multivalued recognition system with an example from biomedical domain to highlight its essence.

3.8 SOLUTIONS/ANSWERS

E1) Use Bayes formula to calculate the a posterior probabilities.

E2) Using the above formula for normal distribution, derive the equation for the posterior probability $P(\omega_i \mid x)$. Calculate then the logarithmic discriminant function $g_i(x) = \log P(\omega_i \mid x)$

Use the result to determine the decision boundary for the classifier. That is, determine x where $g_1(x) = g_2(x)$.

Verify the result using values $\mu_1 = 2$ and $\mu_2 = 4$.

3.9 PRACTICAL ASSIGNMENTS

Session 3

Write a program in C language to construct a function *C = knn(trainclass, traindata, data, k)* that performs k-nearest-neighbour classification. Matrix *traindata* contains training examples so that each column is a single example. Row vector *trainclass* contains the classes of the examples, so that element i of *trainclass* is the class of the example in column i of *traindata*. Matrix data contains samples to be classified, one in each column, and **k** is a parameter which tells number of nearest neighbours used in classification. Return value **C** is a row vector of classes and it should include one value for each column in data. Use Euclidean distance as distance measure.

Algorithm for k-nearest-neighbour classification can be described as follows:

- Find k nearest neighbours from the training set for a sample to be classified.
- Classify the sample to the class which has most training samples among the k nearest neighbours.

Session 4

Write a program in C language to implement a function c = cmeans (data, k) that performs c-means clustering for given data. Matrix data contains training examples so that each column is a single example. Scalar **k** is a parameter which tells the desired number of clusters. Return value c is a matrix which contains means of clusters, one in each column.

One of the simplest, most well-known, and most used clustering techniques is *c-means* which is also known as ISODATA and k-means. The c-means algorithm is as follows:

1. Choose the number of clusters.
2. Initialize the seed vectors for clusters.
3. Cluster the data according to the shortest distance to cluster means.
4. Calculate a mean vector for each cluster and set them as new mean vectors.

5. If there are changes in the mean vectors between iterations i and i-1, then goto 3. Means of clusters $\mu_1, \mu_2, ..., \mu_C$ must have some initial values. These can be obtained by randomly choosing c samples from the data, and assigning $\mu_1, \mu_2, ..., \mu_C$ to the values of these samples.

Session 5

Visualization of C-means clustering

Write a program in C language to modify your cmeans function created in session 4 so that progress of clustering process can be seen. This means that your function should plot original data and original guesses for means of clusters. After each iteration your function should show samples belonging to each cluster with different symbols and/or colors, and your function should also show trajectories of means of clusters starting from initial guesses and ending to final values.

3.10 REFERENCES

1. J. C. Bezdek (1981): "Pattern Recognition with Fuzzy Objective Function Algoritms", Plenum Press, New York.
2. S. Singh and M. Steinl (1996): Fuzzy Search Techniques in Knowledge-Based Systems, Proc. 6th International Conference on Data and Knowledge Systems for Manufacturing and Engineering (DKSME'96), Tempe, Arizona, USA, pp. 1-10.
3. A. Lalandel, L. Legrand, P. Walker, M. Jaulent, F. Guy, Y. Cottin, F. Brunotte (1997): Automatic detection of cardiac contours on MR Images using fuzzy logic and dynamic programming, Proc AMIA Annual Fall Symposium, pp. 474–478.
4. Eric Chen-Kuo Tsao, Jamec C.Bezdek and Nikhil R.Pal (1994): “Fuzzy Kohonen Clustering Networks, Pattern Recognition,” Vol.27, No.5 pp.757-764.
5. S. Dick, S.K. Pal and S. Mitra (1999), Neuro-Fuzzy Pattern Recognition Methods in Soft Computing, IIE Transactions, John Wiley & Sons, New York, 375 pp ., ISBN 0-471-34844-9.