ROUGH CLASSIFICATION IN INFORMATION SYSTEMS UNDER FUZZY ATTRIBUTES

THESIS

Submitted

in partial fulfilment of the requirements for the award of the Degree of

DOCTOR OF PHILOSOPHY

IN THE DEPARTMENT OF COMPUTER APPLICATIONS

By

D. REKHA

Regn. No. SP 10 CA DB 83

DEPARTMENT OF COMPUTER APPLICATIONS

St. PETER'S INSTITUTE OF HIGHER EDUCATION AND RESEARCH

St. PETER'S UNIVERSITY

CHENNAI-600 054

SEPTEMBER 2016

DECLARATION

Certified that the thesis entitled "**Rough Classification in Information Systems under Fuzzy Attributes**" is the bonafide record of independent work done by me under the supervision of **Dr. K. Thangadurai**. Certified further that the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred earlier.

Dr. K. Thangadurai **D. Rekha**

Supervisor

Place : Avadi

Date :

CERTIFICATE

I hereby certify that the thesis entitled, "**Rough Classification of Information Systems under Fuzzy Attributes**" submitted to the St. Peter's University, for the award of Degree of Doctor of Philosophy is the record of research work done by the candidate **D. Rekha** under my guidance and that the thesis has not formed previously the basis for the award of any degree, diploma, associateship, fellowship or other similar titles.

Place : Avadi **Dr. K. Thangadurai**

Date : Supervisor

ACKNOWLEDGEMENTS

It is my foremost duty to express my profound sense of gratitude to my guide **Dr. K. Thangadurai**, Assistant Professor & Head, P.G. & Research Dept. of Computer Science, Government Arts College, Karur for his unwavering support. His wise counsel and valuable suggestions from the budding to the blossoming stage have proved valuable. Not only his valuable guidance but also his unstinted support made this project possible.

I am grateful to **Dr. R. Latha**, Professor, and Head of the Department of Computer Applications, St. Peter's University, Chennai and it is also the prime time to render my sincere and wholehearted thanks to the jury of the doctoral committee members **Dr. P. Sengottuvelan**, Associate Professor, Department of Information Technology, Bannari Amman Institute of Technology, Sathyamangalam, and **Dr. M. Punithavalli**, Director of MCA, Sri Ramakrishna Engineering College, Coimbatore.

My heartfelt thanks are due to **Dr. P. Ponramu**, Assistant Professor in Commerce, Govt College, Melur for helping me in framing and encouraging me during the period of my research.

REKHA D.

ABSTRACT

The Rough set theory was developed by Pawlak in 1982. This theory finds wide applications in knowledge discovery and data mining. Dr. Lofti Zadeh introduced the concept of Fuzzy sets in the year 1965. Fuzzy sets provide the means for representing imprecise data.

As fuzzy logic assumes a non-belonguity measure to be the complement of the belonguity measure, in 1986, Atanassov generalized it into intuitionistic fuzziness. An information system is a relational database in which data are stored as tables. It is defined by a set of attributes some of which are conditional attributes and some decision attributes. Classification is the process of identifying records belonging to different logical groups.

The work deals with classifying the records in an information system in which there are fuzzy or intuitionistic decision attributes. An algorithm for classifying has been designed by using the hedges of the fuzzy / intuitionistic fuzzy attribute, and the approximations using rough sets and threshold variables.

The rough indexing algorithm for the three-way approach is modified and extended the rough indexing to the information systems by incorporating the Probabilistic Naïve Bayesian Rough Set Model. This thesis could be helpful in indexing the various information systems.

In Soft Computing, various types of ambiguity may prevail. Making decisions under these uncertainties is risky. The algorithms and theorems developed in this work would be helpful in developing tools with precision. Using this, a small set of records has been classified based on a decision attribute. The implementation of the algorithm has been done as a console application in Visual C++.

CONTENTS

LIST OF TABLES

FIGURE

CHAPTER 1

INTRODUCTION

1.1 Objective

The elements of Fuzzy sets have degrees of membership. The membership of elements in a set is assessed in binary terms in a classical set theory, according to a bivalent condition – a set either contains or does not contains the element. Lofti Zadeh, the Father of Fuzzy Sets, has defined Soft Computing as a method of computing, different from conventional or hard computing in its tolerance for imprecision, uncertainty, partial truth, and approximation. The role model for Soft Computing is the human mind. The guiding principle of Soft Computing is: "Exploit the resistance for incomplete truth, imprecision, estimation to accomplish tractability strength, low solution cost and uncertainty".

Soft Computing has attracted several researchers and Computer Scientists in developing various technical aspects such as Neural networks, Fuzzy logic, etc. There are several Probabilistic Information Systems and the objective of this work is to provide an indexing technique for the probabilistic information systems with fuzzy attributes.

Fuzzy logic [8] is a type of many-valued logic. It is approximate rather than a fixed and an exact one and deals with reasoning. The truth value of the variables of the Fuzzy logic may range between 0 and 1.The first invention in Soft Computing is Fuzzy Logic. At a later point, the inclusion of other tools has come. The underlying concept of Fuzzy Logic is the Fuzzy Set.

1.2 Review of Literature

In 1965, Zadeh [36] introduced the theory of fuzzy sets. Nowadays fuzzy sets have become an essential tool in developing advanced technologies. Fuzzy sets are based on membership values ranging from 0 to 1. In parallel, fuzzy logic was developed in such a way that it mainly deals with the linguistic variables. Later in 1972, Zadeh introduced the concept of linguistic hedges such as ‘very’, ‘slightly’, etc. Nowadays, fuzzy concepts are involved in several systems nowadays called Fuzzy Systems [37].

Defuzzification is the process of producing a quantifiable result in fuzzy logic, given fuzzy sets and corresponding membership degrees. It is typically needed in fuzzy control systems. For decision making in any fuzzy system, the output is to be de-fuzzified. Some of the methods of defuzzification [28] are a) α-cut b) strong α- cut c) Max Membership Principle. d) Centroid Method e) Weighted Average Method f) Mean-max Membership. In our work, for defuzzification, a strong α-cut is used.

The area in which the straight mathematical modeling fails, the theory of fuzzy sets is used. Z.Pawlak has introduced a useful mathematical tool for Computer Technology which is similar to Fuzzy Sets. The theory of rough sets [18] is involved in several decision-makings, data mining, knowledge representations and knowledge acquisition [41] applications.

In 1981 for information systems [38], Z. Pawlak gave the theoretical foundation. Later, the same approach helped Pawlak to adapt the theory of rough sets. In 1982, he [39] introduced the theory of rough sets. In 1985, he [40] derived the rough dependency theory of attributes in information systems.

A basic problem in many practical applications of the rough sets is defining a method for an efficient selection of the set of attributes (features) necessary for the classification of objects [33] in the considered universe. Such knowledge reduction problems which are the subject of talent are treated in [41]. These knowledge reduction problems are highly involved are common in Information System.

Any information system consists of several attributes. It is tedious to recall each attribute every time. So, it is necessary to avoid the redundant attributes and pick up only the minimal feature. This minimal feature which can be computed using rough sets is called a reduct.

In 1991, Skowron described the method of computing reducts by using discernibility matrices [29]. However, this method cannot list all possible reducts of the information system. In 1999, an algorithm was introduced by Starzyk to list all reducts of the given information system [32].

In recent days, there have been several research works carried out on rough sets under fuzzy environment and they are described in [22, 28]. The present study is an attempt to give an algebraic treatment of the rough and fuzzy approaches.

1.3 Definition of Concepts

1.3.1 Fuzzy Sets

Fuzzy sets are sets whose elements have degrees of membership. The membership of elements in a set is assessed in binary terms in the classical set theory, according to a bivalent condition; a set either contains or does not contain the element. The fuzzy set theory can be used in a wide range of domains in which information is incomplete or imprecise, such as bioinformatics.

1.3.2 Rough Sets

Pawlak has first described the Rough Set, is a formal approximation of a crisp set (i.e., a conventional set) in terms of a pair of sets which provide original set's lower and the upper approximations. As per the standard version of Pawlak's rough set theory, the lower- and upper approximation sets are crisp sets, but in other variations, the approximation sets may be fuzzy sets. Z. Pawlak developed the theory of rough sets in 1982. This theory provides a tool for solving the problems of pattern recognition, knowledge representation, and knowledge acquisition.

1.3.3 Crisp Sets

Crisp sets are the sets that we use the most in our life. An element is either a member of a set or not, in a crisp set. For instance, the class of food known as candy contains a jelly bean, but does not contain the mashed potatoes.

1.3.4 Fuzzy Logic

Fuzzy logic deals with reasoning that is approximate rather than fixed and exact; and is a form of a many-valued logic. Compared to traditional binary sets where variables may take on true or false values, fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. Fuzzy logic has been stretched out to manage the concept of partial truth, where the truth value may go between 'completely true' and 'completely false'.

1.3.5 Information System

The Information system is the investigation of reciprocal systems of equipment and programming that individuals and associations use to gather, channel, handle, make and circulate information. Information Systems incorporate a variety of disciplines such as the analysis and design of systems, computer networking, data security, and database administration and decision supportive systems.

1.3.6 Decision Table

Decision tables are a precise, but compact way to model complicated logic. Decision tables associate conditions with actions to perform, like flowcharts and if-then-else and switch-case statements. They do so in a more elegant way in many cases. Each decision corresponds to a variable, relation or predicate whose possible values are listed among the condition alternatives.

1.3.7 Database Index

A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and storage room to keep up the index data structure. Indexes are utilized to find information rapidly without searching every row in a database table, every time a database table is accessed. Utilizing one or more columns of a database table, the indexes are created, giving the premises for both efficient access of ordered records and rapid random lookups.

1.3.8 Fuzzy Number

A fuzzy number is a generalization of a regular, real number in the sense that it does not refer to one single value, but rather to a connected set of possible values, where each possible value has its own weight between 0 and 1. This weight is called the membership function. Just as Fuzzy logic is an extension of Boolean logic, fuzzy numbers are an extension of real numbers. Calculations with fuzzy numbers allow the incorporation of uncertainty on parameters, properties, geometry, initial conditions, etc.

1.4 Thesis Organization

The **First Chapter** gives an introductory knowledge about various concepts evolved in the objective and explains how Soft Computing, as a method of computing, differs from conventional or hard computing in its tolerance for imprecision, uncertainty, partial truth, and approximation. The guiding principle of Soft Computing is: “Exploit the resistance for incomplete truth, imprecision, estimation to accomplish tractability strength, low solution cost and uncertainty”. It also explains the properties of Fuzzy logic in the respective areas and how it will be used to retrieve the information system.

The **Second Chapter** explains the basic concepts of rough and fuzzy approaches which are essential for this project and also it provide us a deeper understanding of the Crisp Sets and Fuzzy Sets with their definitions, operators that they can handle, Special cases, Standard Norms, Methods of Defuzzification, Fuzzy relations, Approximation, Accuracy and quality of that approximation for the rough sets with their variable precisions.

The **Third Chapter** deals with the introduction of the information system and its types. It defines the architecture of the information system. The

Information system (IS) is the study of complementary networks of hardware and software that individuals and associations use to gather, channel, handle, make and disseminate the information. It also gives knowledge about the Computer information system(s) (CIS), communication technology (ICT) and information which the organization is in need of. Further, this chapter moves on with the Approximation Space, Reduction of Attributes and Data Relation of the Information Systems.

The **Fourth Chapter** discusses the concept of Rough Sets in the information system [27], starts with the Dispensable Indispensable Features, Discernibility Matrix and the Approach for the Elimination Method and various algorithms with their decision attributes examples. It defines an algorithm for reduct generation and introduces the basic concepts of reduct and core.

The **Fifth Chapter** introduces the concept of information language. The information language is used to describe decision rules and decision algorithms in a syntactical way, which allows employing standard logical methods to analyze and investigate. It also provides us the syntax, formulas, properties, decision rules and their algorithms.

The **Sixth Chapter** is focused on the analysis of Fuzzy set using threshold rough set approach [2] on Fuzzy sets, this is defuzzification on rough fuzzy sets and Rough Fuzzy sets have been created. Here, whenever defuzzification is required, a strong α cut is used. R-Domain, the domain of this threshold α has been constructed in order to obey the basic properties of rough sets.

The **Seventh Chapter** deals with the new approach of indexing the similar objects of the information system as with fuzzy decision attributes using the threshold α and finding the lower index of an element using lower approximations. Another algorithm is presented for indexing the values of U by using both the approximations.

The **Eighth Chapter** develops an indexing algorithm on information system with fuzzy decision attributes. The rough indices algorithm is modified for a three-way approach on rough sets. We can find several information systems with fuzzy decision attributes in our real-time systems and consequently the scope of the algorithms discussed would be applicable to such information systems. Here, from the fuzzy decision attribute [19], the Naïve Bayesian Rough Indexing of the data can be derived.

The **Ninth Chapter** deals with rough indexing in the information systems by intuitionistic fuzzy decision attributes and the process is based upon the Probabilistic Naïve Bayesian Rough Set Model. According to the concept of fuzzy sets, the non-membership function gives a value that is equivalent to the variation between the membership value and one. However, this cannot be used in all applications, because some of them defy this non-membership rule. These particular applications prompted Atanassov to develop the concept of intuitionistic fuzzy set. The path he followed allows the dimensional view of fuzzy sets in terms of membership grade and non-membership grade.

The **Tenth Chapter** extends rough indexing to the information systems by incorporating the Probabilistic Naïve Bayesian Rough Set Model and defined the positive, negative and boundary regions by dilating/concentrating the upper and lower threshold values. The same threshold α has been utilized, for several purposes in the above chapters, to make the homogeneity; and we replace the threshold α to obtain a Strong Cut on fuzzy sets with δ in this chapter.

CHAPTER 2

FUZZY SETS, ROUGH SETS

This chapter presents an introduction to crisp sets, fuzzy sets and rough sets in 2.1, 2.2 and 2.3 respectively.

2.1 Crisp Sets

Cantor introduced the theory of crisp sets. Any crisp set can be expressed using its characteristic function. For any Universe of discourse U, the set A is given by the characteristic function χ_A: U→{0,1} defined by

$$\chi_A(x) = \begin{cases} 1 & if \quad x \in A \\ 0 & if \quad x \notin A \end{cases} \tag{2.1}$$

The number of elements in the crisp set A is called the ***cardinality*** or ***power*** of A. It can be mathematically defined as $\sum_{x \in U} \chi_A(x)$.

The concept of crisp sets gives the basics on ***Boolean Logic*** [28]. There are several remarkable theories being constructed based on Boolean Logic.

In 1965, Zadeh introduced the theory of fuzzy sets, which generalizes the theory of crisp sets. This theory is involved in several applications where the Boolean logic fails to give efficiency.

2.2 Fuzzy Sets

Fuzzy sets are obtained by replacing the codomain of the function defined in 2.1 by [0,1]. Here, the function, which is defined, is called ***membership function*** and the value assumed by the membership function is called the grade of membership [12] in the given fuzzy set A. It can be defined as follows:

Consider U={$x_1,x_2,\ldots,x_n$} as the universe of discourse. Then any fuzzy subset A can be defined as $\left\{\frac{\mu_A(x_1)}{x_1}+\frac{\mu_A(x_2)}{x_2}+....+\frac{\mu_A(x_n)}{x_n}\right\}$ where μ_A is the membership function defined from U to [0,1].

As this work focuses on information systems in which the records are invariant, for the computational purpose, the fuzzy set A is represented as $A=(\mu_A(x_1), \mu_A(x_2),\ldots., \mu_A(x_n))$.

2.2.1 Special Cases of Fuzzy Sets

Similar to crisp sets, in fuzzy sets convexity is defined as follows:

Definition:

A ***convex fuzzy set*** [28] is described by a membership function, whose membership values strictly monotonically increases, or strictly monotonically decreases, or whose membership values strictly increases and then strictly monotonically decreases with increasing values for the elements of the universe of discourse.

Definition:

A fuzzy set is said to be ***normal*** [28] if its membership function has at least one element, say x with the membership value 1.

Definition:

If A is a convex single point fuzzy set defined on the real line, then A is called a ***fuzzy number*** [28].

2.2.2. Operators on Fuzzy Sets

For any two fuzzy sets A and B, their union and intersection can be obtained by using the max and min operators, say t-conorms and t-norms. In [28] they are defined as

$\mu_{A\cup B}(x_i)=\max(\mu_A(x_i), \mu_B(x_i))$ and

$\mu_{A\cap B}(x_i)=\min(\mu_A(x_i), \mu_B(x_i))$ respectively.

As in the case of crisp sets, here the union and intersection of two fuzzy sets are not unique i.e., t conorms and t norms are not unique. The basic definitions of t-conorm and t-norm [4] are given below.

Definition:

A t-conorm or S-norm denoted by S, is a binary operation from IxI→ I where I=[0,1] such that for all a,b,c in I, the following properties hold.

a) S(a,1)=1
b) S(a,0)=a
c) S(a,b)=S(b,a)
d) S(a,S(b,c))=S(S(a,b),c)
e) S(a,b) is monotonic in both the variables
f) S is continuous

Some of the standard t- conorms are given below:

Table 2.1 Standard t - conorms

Maximum	Max(a,b)
Lukasiewicz	Min(a+b,1)
Strong	$\begin{cases} \max(a,b) & if \quad \min(a,b)=0 \\ 1 & otherwise \end{cases}$
Probabilistic	a+b-ab
Hamacher	$\frac{a+b-(2-\gamma)ab}{1-(1-\gamma)ab}, \gamma \geq 0$
Yager	$1-\min\left\{1,\left[a^p+b^p\right]^{1/p}\right\}, p>0$

Definition:

A t-norm T is a binary operation from IxI→ I where I=[0,1] such that for all a,b,c in I, the following properties hold.

a) T(a,1)=a
b) T(a,0)=0
c) T(a,b)=T(b,a)
d) T(a,T(b,c))=T(T(a,b),c)
e) T(a,b) is monotonic in both the variables
f) T is continuous

Some of the standard t-norms are given below

Table 2.2 Standard t - norms

Minimum	Min(a,b)
Lukasiewicz	Max(a+b-1,0)
Weak	$\begin{cases} \min(a,b) & if \quad \max(a,b)=1 \\ 0 & otherwise \end{cases}$

Probabilistic	Ab
Hamacher	$\frac{ab}{(\gamma+(1-\gamma)(a+b-ab))}, \gamma \geq 0$
Dubois and Prade	$\frac{ab}{\max\{a,b,\alpha\}}, \alpha \geq 0$
Yager	$1-\min\left\{1,\left[(1-a)^p+(1-b)^p\right]^{1/p}\right\}, p$

However, it is seen that all the norms do not satisfy the entire properties of union and intersection of crisp sets. In addition to the above, the strong α-cut, inclusion and complements of fuzzy sets are defined as follows:

Definition:

A is said to be a subset of B if and only if $\mu_A(x_i) \leq \mu_B(x_i)\ \forall\ x_i \in U$.

Definition:

For any fuzzy set A, the complement of A is given by the grade of each membership element of U; i.e., $\mu_A{}^c(x_i) = 1 - \mu_A(x_i)\ \forall\ x_i \in U$.

As mentioned earlier, in any fuzzy system, it is necessary to de-fuzzify the fuzzy set for any decision-making. There are several tools available for defuzzification.

2.2.3 Methods of Defuzzification

In this section, some of the defuzzification tools are described [28].

2.2.3.1 *Max-membership Principle:*

Also known as the ***height method***, this scheme is limited to peaked output functions. This method is given by

If $\mu_A(z^*) \geq \mu_A(z)$ for all z, then de-fuzzified value of A is z*

2.2.3.2 *Centroid Method:*

It is given by the algebraic expression $z^* = \frac{\int z.\mu_A(z)dz}{\int \mu_A(z)dz}$ where $\int$ denotes an algebraic integration. This procedure is also called ***Centre of the area, Centre of gravity method***.

2.2.3.3. *Weighted Average Method:*

This method is valid only for symmetrical output membership functions. It is given by the algebraic expression $z^* = \frac{\sum z.\mu_A(z)}{\sum \mu_A(z)}$ where Σ denote an algebraic sum.

2.2.3.4 *Mean-Max Membership:*

This method also called the ***middle of maxima*** is closely related to Max Membership Principle, except that the locations of the maximum membership can be non-unique. Here, the arithmetical mean of those locations is considered as the de-fuzzified value.

2.2.3.5 *Centre of Sums:*

This process involves the algebraic sum of the fuzzy sets say A and B, instead of their union. The drawback of this method is that the intersecting areas are added twice. Here, for the collection of fuzzy sets C_1, C_2,…, C_n ,the de-fuzzified value z* is given by $z^* = \frac{\int z\sum_{k=1}^{n} \mu_{C_k}(z)dz}{\int \sum_{k=1}^{n} \mu_{C_k}(z)dz}$. This method is similar to the weighted average method, except at the center of sums method the weights are the areas of the respective membership functions whereas, in the weighted average method, the weights are individual membership values.

2.2.3.6. *Centre of the largest area:*

If the output fuzzy set has at least two convex sub-regions, then the center of gravity of the convex fuzzy sub-region with the largest area is used to obtain the de-fuzzified value z* of the output.

2.2.3.7. *First (or last) of maxima:*

This method uses the overall output or union of all individual output fuzzy sets to determine the smallest value of the domain with a maximized membership degree.

However, the methods discussed from 2.2.3.1 to 2.2.3.7 give a single de-fuzzified value for the given fuzzy sets. There are tools to obtain a set of values under defuzzification. They are given below:

2.2.3.8. *α Cut:*

For a given $\alpha \in (0,1)$, the α-cut of a fuzzy set A is defined as $\{x \in U: \mu_A(x) \geq \alpha\}$.

2.2.3.9 *Strong α Cut:*

For a given $\alpha \in (0,1)$, the strong α-cut of a fuzzy set A is defined as $\{x \in U: \mu_A(x) > \alpha\}$ and is denoted by $A[\alpha]$.

2.2.4 Fuzzy Relations

In an ordinary relation defined from X to Y, it is possible to speak about the existence of any ordered pair of XxY in that relation. But, for a ***fuzzy relation*** [28], the existence of such ordered pairs is ambiguous. Hence, any fuzzy relation R is defined from XxY to [0,1] (symbolically it is denoted as R(X,Y)), where the strength of the mapping is expressed by the membership function of the relation for ordered pairs from two universes or $\mu_R(x,y)$. For example, the relation R= $\left\{\frac{0}{(0,a)}+\frac{0.2}{(0,b)}+\frac{0.5}{(0,c)}+\frac{1}{(1,a)}+\frac{0.3}{(1,b)}+\frac{0.6}{(1,c)}\right\}$ is a fuzzy relation defined from {0,1} to {a,b,c}.

Any fuzzy relation can also be expressed in terms of a matrix. For example, the above fuzzy relation can be expressed as $\begin{bmatrix} 0 & 0.2 & 0.5 \\ 1 & 0.3 & 0.6 \end{bmatrix}$.

The reflexive, symmetric and transitive properties of crisp sets are explained as follows:

Reflexive: The fuzzy relation R(X,X) is said to be reflexive if R(x,x)=1 for all x∈X.

Symmetry: A fuzzy relation R(X,X) is said to be symmetric if R(x,y)=R(y,x) for all x,y∈X.

Transitivity: A fuzzy relation is said to be transitive if $R(x,z) \geq \max_{y \in X} \min[R(x,y), R(y,z)]$

A fuzzy relation that is reflexive, symmetric and transitive is called a ***fuzzy equivalence relation*** *or* ***fuzzy similarity relation*** [22].

The composition of two fuzzy relations is defined by using min, max operators on the corresponding membership values. It can be defined as follows:

2.2.4.1 *Composition of fuzzy relations*

Consider the fuzzy relations R=($\mu_R(x_i, y_j)$)$_{nxm}$ and S=($\mu_S(y_i, z_j)$)$_{mxs}$. Then their composition [22], S°R is given by ($\mu_{S\circ R}(x_i, z_j)$)$_{nxs}$ where

$$\mu_{S\circ R}(x_i, z_j) = \min\{\max(\mu_R(x_i, y_1), \mu_S(y_1, z_j)), \max(\mu_R(x_i, y_2), \mu_S(y_2, z_j)), \ldots, \max(\mu_R(x_i, y_m), \mu_S(y_m, z_j))\}$$

The arithmetical operation *, which is equal to any one of +,-,x,$\div$, between two fuzzy numbers I and J is given by its membership function $\mu_{I*J}(z) = \max_{x*y=z}\{\min(\mu_I(x), \mu_J(y))\}$.

Until now, several mathematical aspects have been built using fuzzy sets. Similar to crisp sets, fuzzy sets have led to a new approach in logic. Based on fuzzy sets, the concept of Fuzzy Logic [24, 28] was introduced and is described below.

2.2.5 Fuzzy Logic

In Boolean logic, a logical proposition is assigned to a crisp set in the universe of discourse. The variables such as ‘beautiful, ‘brilliant’ etc., cannot be used in Boolean logic. Such variables are called the linguistic variables [28]. The linguistic variables take the grades of membership ranging from 0 to 1. The logic developed for processing such linguistic variables is called ***fuzzy logic*** [28]. Fuzzy set theory is the underlying concept of fuzzy logic.

2.2.5.1 *Fuzzy Predicates*

The predicates, which do not have a logical value, are called fuzzy predicates [28]. If p and q are any two fuzzy predicates, with the help of grade of membership, the fuzzy conjunction ($\wedge$), fuzzy disjunction ($\vee$), fuzzy negation (neg), fuzzy implication (→) and fuzzy bi-implication (↔) are defined.

2.2.5.1.1. *Fuzzy Conjunction*

For any two fuzzy predicates p(a) and q(b), its conjunction (∧) is defined by

$$\mu_{(p(a)\wedge q(b))} = \min(\mu_{p(a)}, \mu_{q(b)}) \quad \textbf{(2.2)}$$

2.2.5.1.2. *Fuzzy Disjunction*

For any two fuzzy predicates p(a) and q(b), its disjunction (∧) is defined by

$$\mu_{(p(a)\vee q(b))} = \max(\mu_{p(a)}, \mu_{q(b)}) \quad \textbf{(2.3)}$$

2.2.5.1.3. *Fuzzy Negation*

For a given fuzzy predicate p(a) its negation (neg) is defined by

$$\mu_{(neg(p(a)))} = 1 - \mu_{p(a)} \quad \textbf{(2.4)}$$

2.2.5.1.4. *Fuzzy Implication*

For any two fuzzy predicates p(a) and q(b), the implication p(a) → q(b) is defined by

$$\mu_{(p(a)\rightarrow q(b))} = \max(1 - \mu_{p(a)}, \mu_{q(b)}) \quad \textbf{(2.5)}$$

2.2.5.1.5. *Fuzzy Bi-implication*

For any two fuzzy predicates p(a) and q(b), the bi-implication p(a) ↔ q(b) is defined by

$$\mu_{(p(a)\leftrightarrow q(b))} = \min(\mu_{(p(a)\rightarrow q(b))}, \mu_{(q(b)\rightarrow p(a))}) \quad \textbf{(2.6)}$$

These connectives are illustrated in the following example:

Example:

Let p(x) and q(x) be the fuzzy predicates with the arguments a,b,c and d.

Table 2.3 Fuzzy Logic Connectivities

X	Y	$\mu_{p(x)}$	$\mu_{q(y)}$	$\mu_{p(x)\wedge q(y)}$	$\mu_{p(x)\vee q(y)}$	$\mu_{p(x)\to q(y)}$	$\mu_{p(x)\leftrightarrow q(y)}$	$\mu_{neg\ \ p(x)}$
A	D	0.2	1	0.2	1	1	0.2	0.8
B	a	0.4	0	0	0.4	0.6	0.6	0.6
C	B	0	0.4	0	0.4	0.6	0.6	1
D	c	1	0.5	0.5	1	0.5	0.5	0

However, the generalized fuzzy conjunction and fuzzy disjunction can be obtained from T and S norms. The other generalized connectives are defined as follows:

Definition:

An implication J [4] is a binary operation from IxI→ I in which the following properties hold:

a) $J(p,r)\geq J(q,r)$ if $q\geq p$
b) $J(p,r)\leq J(p,s)$ if $r\leq s$
c) $J(1,t)=t$ for all $t\in I$
d) $J(0,1)=1$ for all $t\in I$
e) $J(p,J(q,r))=J(q,J(p,r))$

Definition:

A strong negation N [4] is a unary operator from I → I such that

a) N is a decreasing function
b) $N(N(a))=a$ for all $a\in I$
c) $N(0)=1$

d) N(1)=0

Definition:

An S implication [4] is obtained from S-norm and a strong negation N as $a \rightarrow b \equiv S(N(a),b)$ for all $a,b \in I$.

In fuzzy logic, with a specific end goal to enhance the effectiveness of fuzziness, the concept of dilation [37] and concentration were introduced. In general, they are called ***hedges***. In 1972, Zadeh introduced this concept of hedges.

Consider the linguistic variable 'low' with the membership function α, for instance. The hedges 'very' and 'very, very' influence the proficiency of the variable with the relating membership values α^2 and α^4. They are called ***concentration*** [37]. The hedges 'slightly' and 'more slightly' influence the proficiency of the linguistic variables with the membership values with the relating membership values $\alpha^{1/2}$ and $\alpha^{1/4}$. They are called ***dilation*** [37].

This work is based on **Rough Sets** apart from fuzzy sets; this chapter gives a brief description of rough sets.

2.3 Rough Sets

In 1982, Z.Pawlak [39] developed the theory of rough sets. This theory provides a tool for solving the problems of pattern recognition, knowledge representation, and knowledge acquisition.

2.3.1. Knowledge base

Consider the finite universe of discourse U and an equivalence relation R on U. Consider the collection U/R of equivalence classes of R. They are referred as ***categories*** or ***concepts*** of R or ***granules*** and $[x]_R$ denotes a category in R containing the element $x \in U$.

A ***knowledge base*** is defined as a relational system K=(U,**Ŕ**), where U is non-empty and **Ŕ** is a family of equivalence relations over U. For any subset P of **Ŕ**, the intersection of all elements of P is also an equivalence relation and is denoted by IND(P) and is called an ***indiscernibility relation*** over P. Moreover, $[x]_{IND(P)} = \bigcap_{R \in P} [x]_R$.

Example:

Consider U={a,b,c,d,e} as the universe of discourse with equivalence relations R and S which produce the equivalence classes {{a,b,c},{d},{e}} and {{a,b,d}, {c,e}} respectively. If P={R,S} then the indiscernibility relation IND(P) partitions U as {{a,b},{c},{d},{e}}.

Here, U/IND(P) denotes the knowledge associated with the family of equivalence relations P, called ***P-basic knowledge*** about U in K.

IND(K) denotes the minimal set of equivalence relations containing all the elementary relations of K. i.e., IND(K)=$\cap${IND(P)/ P$\subseteq$**Ŕ** }

2.3.2. Exact and Rough Sets

Let X be any subset of U. X is said to be ***R-definable*** [41], if X is the union of some R-basic categories. Otherwise, X is ***R-indefinable***. The R-definable sets are also termed as ***R-exact sets*** and R-indefinable sets are termed as ***R-inexact*** or ***R-rough***.

The set X in U is called exact in K if there exists an equivalence relation R in IND (K) such that X is R-exact and X is said to be rough in K if X is R-rough for every R in IND(K).

As some of the sets are rough, it can be explicitly expressed using K. So, it is necessary to approximate them using the elements of K.

2.3.3. Approximations of a set

Let K=(U,**Ŕ**) be a knowledge base and $R \in IND(K)$. Then for any subset X of U, define

$$\underline{R}X=\cup\{Y\in U/R: Y\subseteq X\} \quad \textbf{(2.7)}$$

$$\overline{R}X=\cup\{Y\in U/R: Y\cap X\neq\Phi\} \quad \textbf{(2.8)}$$

Here $\underline{R}X$ and $\overline{R}X$ are said to be R-lower and R-upper approximations of X and ($\underline{R}X$, $\overline{R}X$) is called ***R-rough set*** [41]. If X is R-definable then $\underline{R}X=\overline{R}X$. Otherwise X is R-Rough.

The boundary $BN_R(X)$ is defined as $BN_R(X) = \overline{R}X - \underline{R}X$. Hence, if X is R-definable, then $BN_R(X)=\Phi$. Any object in $\underline{R}X$ gives the certainty of the object in X with respect to R. Any object in $\overline{R}X$ gives the possibility of the object in X with respect to R. Hence, $\underline{R}X$ is called the R-positive region of X and $U-\overline{R}X$ is called the R-negative region of X.

Example:

Consider U={a,b,c,d,e,f} as the universe of discourse and let R be any equivalence relation in IND(K) which partitions U into Ψ ={{a,b,d},{c,f},{e}}. Then for any –subset X={a,b,c,d} of U, $\underline{R}X$={a,b,d} and $\overline{R}X$ ={a,b,c,d,f}. Hence, $BN_R(X)$={c,f}. Hence, the R-positive region of X is {a,b,d} and the R-negative region of X is {e}.

On the other hand, consider a subset Y={c,e,f}. Here, $\underline{R}Y$={c,e,f} and $\overline{R}Y$ ={c,e,f}. Therefore, $BN_R(Y)=\Phi$. Hence, Y is said to be R-definable.

2.3.4. *Properties of Approximations*

The following properties can be observed from the above definitions:

(a) $\underline{R}X \subseteq X \subseteq \overline{R}X$

(b) $\overline{R}(X \cup Y) = \overline{R}X \cup \overline{R}Y$

(c) $\underline{R}(X \cap Y) = \underline{R}X \cap \underline{R}Y$

(d) $\overline{R}(X \cap Y) \subseteq \overline{R}X \cap \overline{R}Y$

(e) $\underline{R}(X \cup Y) \supseteq \underline{R}X \cup \underline{R}Y$

(f) If $X \subseteq Y$ then $\underline{R}X \subseteq \underline{R}Y$; $\overline{R}X \subseteq \overline{R}Y$

(g) $\overline{R}(X^c) = (\underline{R}X)^c$

(h) $\underline{R}(X^c) = (\overline{R}X)^c$

These properties can be illustrated with the following example:

Example:

Consider U={a,b,c,d,e,f,g,h} as the universe of discourse and R be any equivalence relation in IND(K) which partitions U into Ψ ={{a,d,h},{b,e,g},{c},{f}}. Then for any subsets X={a,d,e}, Y={b,h} and Z={a,d} of U, $\underline{R}X=\Phi$, $\underline{R}Y=\Phi$, $\underline{R}Z=\Phi$, $\overline{R}X$ ={a,b,d,e,g,h}, $\overline{R}Y$ ={a,b,d,e,g,h} and $\overline{R}Z$ ={a,d,h}.

a) As $\Phi \subseteq \{a,d,e\} \subseteq \{a,b,d,e,g,h\}$, $\underline{R}X \subseteq X \subseteq \overline{R}X$
b) $X \cup Y = \{a,b,d,e,h\}$ $\Rightarrow$ $\underline{R}(X \cup Y) = \{a,d,h\}$ and $\overline{R}(X \cup Y) = \{a,b,d,e,g,h\}$. Further, $\overline{R}X \cup \overline{R}Y = \{a,b,d,e,g,h\}$, Hence, $\overline{R}(X \cup Y) = \overline{R}X \cup \overline{R}Y$
c) $X \cap Y = \Phi \Rightarrow \underline{R}(X \cap Y) = \Phi$ and $\overline{R}(X \cap Y) = \Phi$. Further, $\underline{R}X \cap \underline{R}Y = \Phi$. Hence, $\underline{R}(X \cap Y) = \underline{R}X \cap \underline{R}Y$
d) $\overline{R}X \cap \overline{R}Y = \{a,b,d,e,g,h\}$ and $\overline{R}(X \cap Y) = \Phi$. Hence, $\overline{R}(X \cap Y) \subseteq \overline{R}X \cap \overline{R}Y$
e) $\underline{R}(X \cup Y) = \{a,d,h\}$ where as $\underline{R}X \cup \underline{R}Y = \Phi$. Hence, $\underline{R}(X \cup Y) \supseteq \underline{R}X \cup \underline{R}Y$
f) This property can be illustrated by taking the sets X and Z.i.e, $X \subseteq Z \Rightarrow \underline{R}X \subseteq \underline{R}Z$; $\overline{R}X \subseteq \overline{R}Z$
g) $X^c = \{b,c,f,g,h\} \Rightarrow \overline{R}(X^c) = \{a,b,c,d,e,f,g,h\}$ which is equal to $(\underline{R}X)^c$
h) $X^c = \{b,c,f,g,h\} \Rightarrow \underline{R}(X^c) = \{c,f\}$. $(\overline{R}X)^c = \{c,f\}$. Hence, they are equal.

In approximation theory, the improper approximation may lead the user to take irrelevant decisions. So, for any approximation, it is necessary to discuss its efficiency.

2.3.5. Accuracy and quality of approximation

As the inexactness depends on the boundary, the ***accuracy*** [41] of the approximation of a nonempty set X can be defined as follows:

$$\alpha_R(X) = \frac{card\,\underline{R}X}{card\,\overline{R}X} \qquad \textbf{(2.9)}$$

It can be observed that whenever the set is exact, $\alpha_R(X)$ is 1 and in general $0 \le \alpha_R(X) \le 1$. Using the accuracy, another measure called ***R-roughness*** of X, can be defined as $\rho_R(X) = 1 - \alpha_R(X)$.

The above definition can be extended to the families of subsets of U. Consider the family $F = \{X_1, X_2, \ldots, X_n\}$ of nonempty subsets of U. Then the lower and upper approximations $\underline{R}F$ and $\overline{R}F$ of the family F are given by $\underline{R}F=\{\underline{R}X_1,\underline{R}X_2,\ldots,\underline{R}X_n\}$ and $\overline{R}F=\{\overline{R}X_1, \overline{R}X_2,\ldots, \overline{R}X_n\}$ respectively. The ***accuracy of approximation*** [41] of F by R is given by

$$\alpha_R(F) = \frac{\sum card\,\underline{R}X_i}{\sum card\,\overline{R}X_i} \qquad \textbf{(2.10)}$$

The second measure is called the ***quality of approximation*** [41] of F by R, which is given by

$$\gamma_R(F) = \frac{\sum card\,\underline{R}X_i}{CardU} \qquad \textbf{(2.11)}$$

This theory has attracted several researchers to apply approximations on equality and inclusion of sets.

2.3.6. Rough equality and inclusion of sets

This section deals with hybridizing rough set concepts with basic set equality and inclusion.

2.3.6.1. *Rough equality of Sets*

Two sets X and Y are said to be ***bottom R-equal*** [41], if they have the same R-lower approximations. If they have the same R-upper approximations they are called ***top R-equal*** [41]. The sets are said to be ***R- equal*** [41], if they are top and bottom R-equal. The bottom R-equalness, top R-equalness, and R-equalness are denoted by $\eqsim_R$, $\simeq_R$ and $\approx_R$ respectively.

The following properties can be observed from the above definitions.

(a) $X \eqsim Y$ iff $X \cap Y \eqsim X$ and $X \cap Y \eqsim Y$

(b) $X \simeq Y$ iff $X \cup Y \simeq X$ and $X \cup Y \simeq Y$

(c) If $X \simeq X'$ and $Y \simeq Y'$ then $X \cup Y \simeq X' \cup Y'$

(d) If $X \approx X'$ and $Y \approx Y'$ then $X \cap Y \approx X' \cap Y'$

(e) If $X \simeq Y$ then $X \cup Y^c \simeq U$

(f) If $X \approx Y$ then $X \cap Y^c \approx \Phi$

(g) If $X \subseteq Y$ and $Y \simeq \Phi$ then $X \simeq \Phi$

(h) If $X \subseteq Y$ and $X \approx U$ then $Y \approx U$

(i) $X \simeq Y$ iff $X^c \approx Y^c$

(j) If $X \approx \Phi$ or $Y \approx \Phi$ then $X \cap Y \approx \Phi$

(k) If $X \simeq U$ or $Y \simeq U$ then $X \cup Y \simeq U$

2.3.6.2 *Rough Inclusion of Sets*

If $\underline{R}X \subseteq \underline{R}Y$, then X is said to be ***bottom R included*** in Y [41] and is denoted by $X \subseteq_R Y$. If $\overline{R}X \subseteq \overline{R}Y$ then X is said to be ***top R included*** in Y [41] and is denoted by $X \overline{\subset}_R Y$. If X is bottom and top R included in Y, then X is said to be ***R-included*** in Y and is denoted as $X \underline{\overline{\subset}}_R Y$.

As this entire work does not involve any computation using indiscernibility relations, for convenience, the notations $\overline{A}$ and $\underline{A}$ are used for $\overline{R}A$ and $\underline{R}A$ respectively.

2.3.7 Variable Precision Rough Sets

The cardinal limitation of the rough set theory is classification analysis. The other major limitation originates from the assumption that the universe U of data objects is known and all the conclusions derived from the model are applicable to this set of objects. These limitations are checked by using a technique obtained from the extended model of rough sets that is Variable Precision Rough Set model [VPRS] which was proposed by W.Ziarko in 1993 [42].

Variable precision rough set model is developed with the statistical approach because of which set-theoretic properties of rough sets do not find a place in this concept.

Suppose X and Y are any two subsets of the universe of discourse U, then define the measure c(X,Y), the relative degree of misclassification of the set X with respect to Y as:

$$c(X,Y) = \begin{cases} 1 - \dfrac{|X \cap Y|}{|X|} & if \quad |X| > 0 \\ 0 & if \quad |X| = 0 \end{cases} \qquad \textbf{(2.12)}$$

In order to classify all the elements of the set X into the set Y, c(X,Y)*100% is called the classification error. Also, it is to be noted that the product c(X,Y)* |X| gives the number of misclassified elements known as absolute classification errors. Clearly $Y \supseteq X$ if and only if c(X,Y)=0.

2.3.7.1 *Majority inclusion*

For $0 \leq \beta < 0.5$, the majority inclusion is defined as $Y \overset{\beta}{\supseteq} X$ if and only if $c(X,Y) \leq \beta$

***2.3.7.2** Approximations of a variable precision rough set model*

For any set $X \subseteq U$ of the universe of discourse, the generalized β-lower approximation and β-upper approximation are obtained by replacing inclusion relation with majority inclusion and these are given by the following equations:

$$\underline{R}_{\beta} X = \cup\{E \in R^* : X \overset{\beta}{\supseteq} E\} \text{ (OR)}$$
$$\underline{R}_{\beta} X = \cup\{E \in R^* : c(E, X) \leq \beta\} \text{ and}$$
$$\overline{R}_{\beta} X = \cup\{E \in R^* : c(E, X) < 1 - \beta\} \qquad \textbf{(2.13)}$$

The boundary region of a set is given by $BNR_{\beta} X = \cup\{E \in R^* : \beta < c(E, X) < 1 - \beta\}$.The β-lower approximation is also called β-positive region and β-negative region of X is defined as a complement of the β-upper approximation i.e.,

$$NEGR_{\beta} X = \cup\{E \in R^* : c(E, X) \geq 1 - \beta\} \qquad \textbf{(2.14)}$$

Example:

Consider the approximation space A=(U,R) with U={a,b,c,d,e,f,g} and the equivalence classes of the relation R is given by E_1={a,b,d}, E_2={c,e,f}, E_3={g} and X={a,c,e} be any subset of U. Then

$$c(E_1, X) = 1 - \frac{|\{a,b,d\} \cap \{a,c,e\}|}{|\{a,b,d\}|} = 1 - \frac{1}{3} = 0.66$$

$$c(E_2, X) = 1 - \frac{|\{c,e,f\} \cap \{a,c,e\}|}{|\{c,e,f\}|} = 1 - \frac{2}{3} = 0.33$$

$$c(E_3, X) = 1 - \frac{|\{g\} \cap \{a,c,e\}|}{|\{g\}|} = 1 - \frac{0}{1} = 1.$$

If $\beta = 0.4$ then by definition, β-lower approximation={c,e,f}, β-upper approximation={c,e,f}, β-boundary region=Φ and β-negative region ={a,b,d,g}.

2.3.7.2.1 *Properties of β-approximations*

The following properties are satisfied by the β-approximations for every $0 \le \beta < 0.5$

1) (a) $X \overset{\beta}{\supseteq} \underline{R}_{\beta} X$

(b) $\overline{R}_{\beta} X \supseteq \underline{R}_{\beta} X$

2) $\underline{R}_{\beta}\Phi = \overline{R}_{\beta}\Phi = \Phi$; $\underline{R}_{\beta}U = \overline{R}_{\beta}U = U$

3) $\overline{R}_{\beta}(X \cup Y) \supseteq \overline{R}_{\beta} X \cup \overline{R}_{\beta} Y$

4) $\underline{R}_{\beta} X \cap \underline{R}_{\beta} Y \supseteq \underline{R}_{\beta}(X \cap Y)$

5) $\underline{R}_{\beta}(X \cup Y) \supseteq \underline{R}_{\beta} X \cup \underline{R}_{\beta} Y$

6) $\overline{R}_{\beta} X \cap \overline{R}_{\beta} Y \supseteq \overline{R}_{\beta}(X \cap Y)$

7) $\underline{R}_{\beta}(X^{c}) = (\overline{R}_{\beta}(X))^{c}$

8) $\overline{R}_{\beta}(X^{c}) = (\underline{R}_{\beta} X)^{c}$

Example:

Consider the approximation space A=(U,R) with U={a,b,c,d,e,f,g} and the equivalence classes of the relation R is given by E_1={a,b,d}, E_2={c,e,f}, E_3={g}. Let X={c,d,e,f} and Y={a,b} are any subsets of U. X∪Y={a,b,c,d,e,f}, X∩Y=Φ and X^c={a,b,g}. The relative degree of misclassifications of the sets E_1,E_2 and E_3 is given by c(E_1,X)=0.66, c(E_2,X)=0, c(E_3,X)=1, c(E_1,Y)=0.33, c(E_2,Y)=1, c(E_3,Y)=1, c(E_1,X∪Y)=0, c(E_2,X∪Y)=0, c(E_3,X∪Y)=1, c(E_1,X∩Y)= c(E_2,X∩Y)= c(E_3,X∩Y)=1, c(E_1,X^c)=0.33, c(E_2,X^c)=1 and

$c(E_3,X^c)$=0. If β=0.3, then $\underline{R}_\beta X$ ={c,e,f}; $\overline{R}_\beta X$ ={a,b,c,d,e,f}; $\underline{R}_\beta Y = \Phi$; $\overline{R}_\beta Y$ ={a,b,d} $\underline{R}_\beta (X \cup Y) = \overline{R}_\beta (X \cup Y)$ ={a,b,c,d,e,f}; $\underline{R}_\beta (X \cap Y) = \overline{R}_\beta (X \cap Y) = \Phi$; $\underline{R}_\beta (X^c)$ ={g} and $\overline{R}_\beta (X^c)$ ={a,b,d,g}. Then

1) (a) $c(\underline{R}_\beta X, X) = 1 - \dfrac{|\{c,e,f\} \cap \{c,d,e,f\}|}{|\{c,e,f\}|} = 1 - \dfrac{3}{3} = 0$

Since β=0.3 implies $c(\underline{R}_\beta X, X)$ =0<0.3. Therefore, $X \overset{\beta}{\supseteq} \underline{R}_\beta X$

(b) Since {a,b,c,d,e,f}⊇{c,e,f}, hence $\overline{R}_\beta X \supseteq \underline{R}_\beta X$

2) Clearly, $\underline{R}_\beta \Phi = \overline{R}_\beta \Phi = \Phi$; $\underline{R}_\beta U = \overline{R}_\beta U = U$

3) $\overline{R}_\beta X \cup \overline{R}_\beta Y$ ={ a,b,c,d,e,f}= $\overline{R}_\beta (X \cup Y)$

4) $\underline{R}_\beta X \cap \underline{R}_\beta Y = \Phi = \underline{R}_\beta (X \cap Y)$

5) $\underline{R}_\beta X \cup \underline{R}_\beta Y$ ={c,e,f}⊆{a,b,c,d,e,f}= $\underline{R}_\beta (X \cup Y)$ .Hence $\underline{R}_\beta (X \cup Y) \supseteq \underline{R}_\beta X \cup \underline{R}_\beta Y$

6) $\overline{R}_\beta X \cap \overline{R}_\beta Y$ = {a,b,d}⊇Φ= $\overline{R}_\beta (X \cap Y)$

7) $(\overline{R}_\beta (X))^c$ = {g} = $\underline{R}_\beta (X^c)$

8) $(\underline{R}_\beta X)^c$ = {a,b,d,g}= $\overline{R}_\beta (X^c)$

In order to adopt the approach of rough sets for various applications, mathematicians and computer researchers have generalized rough sets in different directions.

2.3.8. Generalized Rough Sets

Several methods are proposed for [34] generalizing the theory of rough sets. One of the widely used methods is described below:

It is known that the concept of rough sets is dealt with based on the equivalence relations. In generalized rough sets, the equivalence relation conditions are relaxed by tolerance relations. i.e., the reflexive and symmetric relations are considered and the transitive rule is relaxed. This process yields the knowledge base with overlapping granules. For the tolerance relation E, denote $[x]_E$ as the granule containing x. Here, it is to be noted that there may be more than one granule, containing x. Then for any set A of U, the approximations are given by

$$\underline{A} = \cup\{[x]_E \,/\, [x]_E \subseteq A\} \quad \textbf{(2.15)}$$

$$\overline{A} = \cup\{[x]_E / \, [x]_E \cap A \neq \Phi\} \quad \textbf{(2.16)}$$

In this chapter, the basic concepts of rough and fuzzy concepts have been dealt. By the definition of rough sets, one may observe that the region between positive and negative regions is ambiguous.

CHAPTER 3

INFORMATION SYSTEMS

3.1 Introduction

Information system (IS) is the study of complementary networks of hardware and software that individuals and associations use to gather, channel, handle, make and disseminate the information. Information Systems incorporate a variety of disciplines such as the analysis and design of systems, computer networking, data security, database administration and decision supportive systems. Information Management manages the functional and speculative problems of gathering and examining information in a business practical locale which includes the electronic trade, business profitability instruments, applications programming and usage, information mining, decision support and advanced media generation. The telecommunication advancements are managed by Communications and Networking. Information System spans business and computer science utilizing the theoretical establishments of information and computation to study various business models and related algorithmic processes within a computer science discipline. Computer information system(s) (CIS) is a field in which computers and algorithmic processes, including their principles, their software and hardware designs, their applications, and their effect on society are concentrated on while IS underlines usefulness over configuration.

A particular information system plans to support decision making, adminstration, and operations. In an expensive sense, the term is utilized to refer not only to the information and communication technology (ICT) that an organization can use, and also, in support of business procedures to interact with this technology. Information systems additionally can be characterized as a collection of hardware, software, information, people and methods that work together to create the quality data.

3.2 Types of Information Systems

The "work of art" perspective of Information systems found in 1980s was a pyramid of frameworks that clearly explains the hierarchy of the organization, the transaction processing systems at the base of the pyramid, extended by decision supportive networks, administrative information systems, ending with the executive information systems at the top. Despite the fact that the pyramid model stays valuable since it was initially planned various new technologies have been created and new classes of information systems have raised, some of which, no more, fit effectively into the first pyramid model.

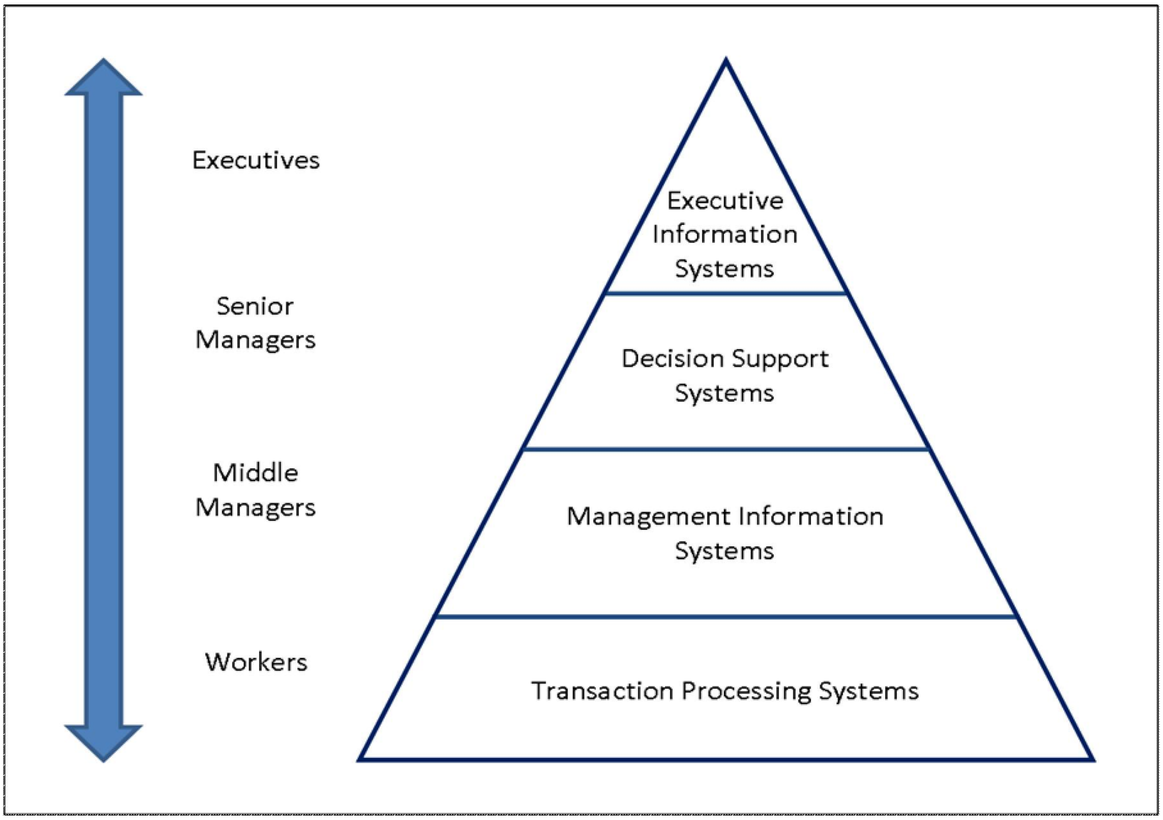

Figure 3.1 Types of Information System

Few samples of such systems are given below:

a) Data warehouses
b) Enterprise resource planning
c) Enterprise frameworks
d) Expert frameworks
e) Search engines
f) Geographical Information system
g) Global data framework and
h) Office computerization.

3.3 Computer Based Information Systems

The fundamental segments of computer-based information system are:

a) ***Hardware-*** these are the devices like the monitor, processor, printer, and keyboard, all of which cooperate together to acknowledge, process, show information and data.
b) ***Software-*** is a program that permits the hardware to process the information.
c) ***Databases-*** are the social event of related documents or tables containing related information.
d) ***Networks-*** are a connecting system that permits the diverse computers for distributing the resources.
e) ***Procedures-*** are the orders for consolidating the parts above to process data and produce the favored yield.

The initial four parts (database, hardware, network and software) forms the term information technology platform. Information technology workers could then utilize these parts to create information systems that watch over security administration of information, measures, and the risks. Such activities are termed as information technology services.

3.4 Information System and Approximation Space

An information system is a 4-tuple

S = (Univ, Att, Val, f),

where

Univ - is a finite set of objects - called the universe,

Att - is a finite set of attributes, Val = $\bigcup_{a \in Att} Val_a$, Val_a – is a finite set of values of $a \in Att$

f : Univ × Att → Val – is a total function, such that

f (x, a) ∈ Val_a for every $a \in Att$ and x ∈ Univ

In other words, an information system is a finite table column of which are labeled by attributes and rows are labeled by objects. Each entry of column a and row x is the value of f(x,a).

An example of an information system is given in the table below:-

Table 3.1 Information System Example

Univ	A	B	c
x_1	1	0	2
x_2	0	1	1
x_3	2	0	0
x_4	1	0	2
x_5	1	0	0
x_6	0	1	1
x_7	2	0	0
x_8	1	0	0

x_9	1	0	2
x_{10}	0	1	1

With every subset of attributes, A $\subseteq Att$, we associate an indiscernibility relation $\underset{A}{\sim}$ defined thus

$$\underset{A}{\sim} = \{(x,y): f(x,a) = f(y,a) \text{ for every } a \in A. \qquad \textbf{(3.1)}$$

Of course $\underset{A}{\sim}$ is an equivalence relation. Hence with every information system S, we can associate an approximation space [10] A_s = (Univ, Ind) where

Ind = { $\underset{a}{\sim}$: a ∈ Att }

Thus every a is a primitive indiscernibility relation in Apr for any Att. In what follows we shall write A instead $\underset{A}{\sim}$ if there will be no confusion of the set attributes of A and the relation $\underset{A}{\sim}$.

Conversely, with every approximation space Apr =(Univ, Ind) we can associate an information system S_{apr} defined as follows:

> With every primitive indiscernibility relation R in Apr, we associate a unique name a_R of R and we define the function f in such a way that f(x, a_R) = f(y, a_R)iff x and y belong to the same equivalence class of the relation R.

Thus the concepts of an approximation space and information space are isomorphic and can be mutually replaced.

3.5 Dependency of Attributes in Information Systems

Let S= {Univ, Att, Val, f} be an information system and A, B⊆ Att subset of attributes. Set of attributes B depends on set of attributes B in a degree k ($0 \leq k \leq 1$) in S, in symbols A $\underset{k}{\Rightarrow}$ B, if k = $\gamma_A(B) = \frac{card\ Int_{A(B^*)}}{card\ Univ}$

The intuitive meaning of this definition is as following: if A $\underset{1}{\Rightarrow}$ B that is to mean that the values of attributes b $\in B$ can be determined when values of attributes a $\in A$ are known. If k ≠ 1, that is to mean that the dependency between A and B is partial, i.e. the dependency holds for objects belonging to $Int_A(B^*)$

For example, in the information system, the following dependency is valid: B $\underset{0}{\Rightarrow}$.5 c, where B ={a,b}, and c is the abbreviation for $\tilde{c}$.

Because

$$c^* = \{ X_1 , X_2 , X_3 \},$$

where

$$X_1 = \{ x_1, x_5, x_9 \},$$
$$X_2 = \{ x_2, x_7, x_{10} \},$$
$$X_3 = \{ x_3, x_4, x_6 , x_8 \}$$

and

$$B^* = \{ Y_1 , Y_2 , Y_3 \},$$

Where

$$Y_1 = \{ x_1, x_4, x_5, x_8, x_9 \},$$
$$Y_2 = \{ x_2, x_7, x_{10} \},$$
$$Y_3 = \{ x_3, x_6 \}$$

Hence

$$\underline{B}X_1 = \emptyset,\ \underline{B}X_2 = Y_2, \underline{B}X_3 = Y_3$$

and

$$Int_B(c^*) = \underline{B}X_1\ U\ \underline{B}X_2\ U\ \underline{B}X_3 = Y_1\ U\ Y_3 = \{x_2, x_3, x_6, x_7, x_{10}\}$$

Thus k = 5/10 = 0.5

This is to mean that only the objects $\{x_2, x_3, x_6, x_7, x_{10}\}$ only can be property classified, to the classes of c^*, employing the set of attributes B. Let us also notice that

$$\bar{B}X_1 = Y_1,\ \bar{B}X_2 = Y_2, \bar{B}X_3 = Y_1\ U\ Y_3$$

i.e. if set X_1 is internally B- discernible, X_2 is B-discernible and X_3 is roughly B- discernible, the corresponding accuracy coefficients are:

$$\alpha_B(X_1) = \frac{0}{5} = 0,$$
$$\alpha_B(X_2) = 3/3 = 1,$$
$$\alpha_B(X_3) = 2/7$$

3.6 Reduction of Attributes

Let B = (Univ, Att, Val, f) be an information system, A, B, C ⊆ Att subsets of attributes. If A is a reduct of B with respect to C, which is to mean that set of attributes A⊆ B provides the same discernibility of classes of the classification C^* as the set of attributes B.

Hence instead using a set of attributes B we can use a smaller set of attributes A, to classify objects to classes of the classification C^*.

It is easy to see, for example, that the information system shown in the table 3.4.1, set {a,c} is the only reduct of set of attributes {a,b,c}, for {a,c} = {a,b,c}. There are two reducts a and c of the set attributes {a,c} with respect to b. Note that the b-core of {a,c} is the empty set. Moreover we have a→ a and c → b in the information system considered.

3.7 Information Relations Derived from Information Systems

Let an information system (OB, AT, {Val_a : a∈ AT} be given. Relationships among the objects from set OB are determined by their properties. Typically, the relationships have the form of binary relations. These relations are referred to as information relations. There are two major groups of information relations: the relations that reflect various forms of indistinguishability of objects in terms of their properties and the relations that indicate the distinguishing ability of the objects. Below, we present a list of the classes of atomic relations that generate the whole family of information relations. To each of these classes, we assign a name that suggests an aspect of the incompleteness of information that the underlying relations reflect.

3.7.1 Indistinguishability Relations

a) Strong (weak) indiscernibility: (x,y) ∈ ind(A) (wind(A)) iff a(x) = a(y) for all (some) a ∈ A.

b) Strong (weak) similarity: (x,y) ∈ sim(A) (wsim(A)) iff a(x) ∩ a(y) ≠ Ø for all (some) a ∈ A.

c) Strong (weak) forward inclusion: (x,y) ∈ fin (wfin(A)) iff a(x) ⊆ a(y) for all (some) a ∈ A.

d) Strong (weak) backward inclusion: $(x,y) \in bin$ (wbin(A)) iff $a(y) \subseteq a(x)$ for all (some) $a \in A$.

e) Strong (weak) negative similarity: $(x,y) \in nim(A)$ (wnim(A)) iff $-a(x) \cap -a(y) \neq \emptyset$ for all (some) $a \in A$.

f) Strong (weak) incomplementarity: $(x,y) \in icom(A)$ (wicom(A)) $a(x) \neq -a(y)$ for all (some) $a \in A$.

CHAPTER 4

ROUGH SETS IN INFORMATION SYSTEMS

The decision table of any information system [41] is given by T=(U, A, C, D) in the theory of rough sets, where U is the universe of discourse, A is a set of primitive features, the subset of A, 'C' is called conditional and the subset of A, 'D' is called the decisional feature respectively.

A binary relation IND (P), called the indiscernibility relation for any subset P of A is defined as IND (P)={(x,y)∈UxU : a(x)=a(y) for all a in P}

We signify the classes obtained by the relation IND (P) by U/IND (P) or U/P. The lower and upper approximations for the indiscernibility relation IND(R) are defined as

$$\underline{R}X = \cup\{Y \in U/R : Y \subseteq X\} \text{ and} \tag{4.1}$$

$$\overline{R}X = \cup\{Y \in U/R : Y \cap X \neq \Phi\} \text{ respectively.} \tag{4.2}$$

The classes U/IND(C) and U/IND (D) are termed as condition and decision classes respectively.

The C-Positive region of D is given by $POS_C(D) = \bigcup_{X \in U/D} \underline{C}X$.

4.1 Dispensable and indispensable Features

Consider c∈C. a feature c is dispensable [25, 41] in T, if $POS_{C-\{c\}}(D) = POS_C(D)$; else the feature c is called indispensable in T. By removing c from T, if c is indispensable, makes T be inconsistent. If all the features of T are indispensable, then T is said to be independent.

4.1.1 Reduct and Core

A reduct is termed as a set of features R in C, if T'=(U, A, R, D) is independent and $POS_R(D)=POS_C(D)$. A reduct can also be termed as the minimal feature subset [13, 41] preserving the above condition.

CORE(C) [13, 41] is defined as the set of all features indispensable in C. As such, CORE(C) = $\cap$RED(C) where RED(C) is termed as the set of all reducts of C.

4.1.2 Discernibility Matrix

The representation of the decision table into discernibility matrix to compute reduct was described by .Skowron [29]. Consider T=(U,A,C,D) be a decision table, with $U=\{x_1,x_2,\ldots,x_n\}$. By a discernibility matrix of T, denoted M(T), n x n matrix defined as

$$m_{ij}=\{a\in C : a(x_i)\neq a(x_j) \quad \wedge (d\in D,\ d(x_i)\neq d(x_j))\} \text{ for } i,j=1,2,\ldots,n \qquad \textbf{(4.3)}$$

Two attributes 'x' and 'y' are said to be strongly equivalent if they appear always together with the elements of the discernibility matrix in the decision table. Each element can be viewed as the disjunctive expression. i.e., if an element of the discernibility matrix is {a,b,c} then it can be noticed as $a\vee b\vee c$. By taking the conjunction of the disjunctive expressions of the discernibility matrix, the discernibility function is provided.

***Example*:**

Given below is the knowledge representation system with C={a,b,c,d} and D={E}.

Table 3.2 Knowledge Representation

	A	b	c	D	E
x_1	1	0	2	1	1
x_2	1	0	2	0	1
x_3	1	2	0	0	2
x_4	1	2	2	1	0
x_5	2	1	0	0	2
x_6	2	1	1	0	2
x_7	2	1	2	1	1

The discernibility matrix is given below as

Table 3.3 Discernibility Matrix

	x_1	x_2	x_3	x_4	x_5	x_6
x_2	---					
x_3	b,c,d	b,c				
x_4	B	b,d	c,d			
x_5	a,b,c,d	a,b,c	---	a,b,c,d		
x_6	a,b,c,d	a,b,c	---	a,b,c,d	---	
x_7	---	---	a,b,c,d	a,b	c,d	c,d

The reducts of the decision table can be found using the discernibility matrix; and it is explained in the section 4.1.4.

4.1.3 Core and Reducts through Discernibility matrix

A set of all singleton entries in the discernibility matrix is termed as the core. The minimal element [29, 15] in the discernibility matrix, which intersects all the elements of the discernibility matrix, is termed as the reduct.

In the discernibility matrix given in 4.1.3, the reducts are {b,c} and {b,d}; and the core is {b}.

In real situations, sometimes it may not be appropriate for granulation [41] of the knowledge base into partitions. For example, in medical diagnosis, the symptoms cannot be partitioned using diseases. In 1996, Yao described the generalization of rough set theory.

The reducts are helpful in characterizing the data in any information system. In general, by using discernibility matrices, only a few reducts can be found. Hence, sometimes it is difficult to list all the reducts. An algorithm for computing reducts using strong equivalence and the law of expansion on the data was developed by Janusz Starzyk [32]. Despite, the execution of this algorithm is cumbersome for the huge volume of data. This chapter explains a technique for obtaining the reduct of the entire system by partitioning it into two with respect to records and obtaining the reducts of the two subsystems and the '***between reducts***'. Moreover, it also explains a technique for consolidating the reducts computed at the clients to get the global reducts.

4.2 An Approach to Reduct Generation Algorithm

Initially, it is essential to discuss the elimination method [32], which is helpful in finding the reducts. In this method, it is essential to verify all the possible combinations of data to find the reduct. Thus, for a system with a limited number of records and attributes, it is effective.

4.2.1. Elimination method

The first step in generating reducts is to make the information system non-ambiguous and eliminating duplicates. The information system is said to be ambiguous when two records are identical with respect to conditional attributes which differ in decisional attributes. When this happens, both the records are to be removed from the information system. The procedure from this point is to take all possible combinations of the conditional attributes. Each combination forms a set, which is checked for ambiguity. If the set of attributes has no uncertainty, then the set is called a reduct. This process is termed as the Elimination method.

As the elimination method is limited as mentioned above, in general, the Reduct generation algorithm is used to compute reducts. Keeping in mind the end goal to continue further, it is important to know the expansion law, which is utilized as part of the algorithm. Here, the elements of the discernibility matrix are noted in OR form. For instance, the element {b,c,d} is viewed as $b \vee c \vee d$. Further, using the connective AND, the entire matrix can be composed.

Here, one may misdirect the above, by treating AND and OR with usual conjunction and disjunction. As the attributes are not 0 or 1, they are to be seen as absent or present accordingly.

4.2.2 Expansion Law

The procedure for expansion law is described as algorithm given below.

1. Discover the attribute X that occurs mostly(at least twice)

2. Apply AND of X and all other OR forms of the elements of the discernibility matrix which do not contain X
3. Apply the connective AND between the OR form of all the elements, in which if the element contains X, remove X.
4. Combine the elements obtained from (2) and (3) by AND.

Example:

Let us consider the elements of the discernibility matrix {{a,b,e},{a,b},{a,c},{d}}. The discernibility relation is defined by $(a \vee b \vee e) \wedge (a \vee b) \wedge (a \vee c) \wedge d$

The element ‘a’ occurs frequently here.

On applying AND `a’ with `d’, $\{a\} \wedge \{d\} = \{a,d\}$, say as component 1

On applying AND for $b \vee e$, b and c, $(b \vee e) \wedge (b) \wedge (c) = \{b, c\} \wedge \{b, c, e\}$, say as component 2

The Integrated form is {a, d}, {b,c},{b,c,e}

4.2.3. Reduct Generation Algorithm

In this section, we discuss the algorithm for reduct generation which is proposed by Janusz Starzyk, Dale E.Nelson and Kirk Sturtz [32].

4.2.3.1 *Algorithm*

Given $f = f_1 \wedge f_2 \wedge \ldots \wedge f_t$ is the discernibility function

Step 1: Apply absorption law to eliminate all disjunctive expressions, which are supersets of another disjunctive expression.

Step 2: Replace each set of strongly equivalent attributes with a dummy variable [the strong equivalence between any two elements a and b is given by $a \in A$ if and only if $b \in A$].

Step 3: Select the attribute, which belongs to a large number of conjunctive sets, numbering at least two, and apply the expansion law.

Step 4: Repeat steps 1 to 3 until the expansion law cannot be applied to each component.

Step 5: Substitute all strongly equivalent classes with their corresponding attributes.

Step 6: Calculate the reducts in every component.

Step 7: Write the Integrated reduct.

The above algorithm is illustrated in the following example:

Example:

Consider the discernibility relation $F=\{a \vee b \vee c \vee f\} \wedge \{ b \vee d\} \wedge \{ a \vee d \vee e \vee f\} \wedge \{ a \vee b \vee c \vee d\} \wedge \{ b \vee d \vee e \vee f\} \wedge \{ c \vee d\}$.

On applying absorption law, as $\{ b \vee d\} \subseteq \{ a \vee b \vee c \vee d\}$, $\{ b \vee d\} \wedge \{a \vee b \vee c \vee d\}=\{ b \vee d\}$. Similarly, $\{ b \vee d \vee e \vee f\} \wedge \{ b \vee d\} =\{ b \vee d\}$. Hence, the discernibility relation becomes $F=\{a \vee b \vee c \vee f\} \wedge \{ b \vee d\} \wedge \{ a \vee d \vee e \vee f\} \wedge \{ c \vee d\}$.

It is observed that {a,f} are strongly equivalent. Denote $a \vee f=M$. Hence, the discernibility relation becomes $F=\{M \vee b \vee c\} \wedge \{b \vee d\} \wedge \{M \vee d \vee e\} \wedge \{c \vee d\}$.

The attribute'd' appears most frequently. Using it apply the expansion law: $F=[\{d\} \wedge \{M \vee b \vee c\}] \wedge [\{M \vee b \vee c\} \wedge \{b\} \wedge \{M \vee e\} \wedge \{c\}]$. By applying the absorption law in component 2, $F=[\{d\}\wedge\{M \vee b \vee c\}]\wedge [\{b\} \wedge \{M \vee e\} \wedge \{c\}]$.

Now all the components are in a simple form.

On replacing M by $a\vee f$, $F=[\{d\}\wedge\{ a \vee f \vee b \vee c\}]\wedge [\{b\} \wedge \{ a \vee f \vee e\} \wedge \{c\}]$.

The reduct of the first component is {a,d},{d,f},{b,d},{c,d} and the reduct of the second component is {a,b,c}{b,c,f},{b,c,e}.

Hence, the Integrated reduct is {a,d},{d,f},{b,d},{c,d}, {a,b,c}{b,c,f},{b,c,e}

In the above example, the method of computing reducts was illustrated. As the information system can be a relational database [7], sometimes there may be a necessity for combining the reducts from two or more clients to get the global reduct. For instance, about the decision on some vital issues, if they are sent to many referees from the server and if one receives a set of all reducts from each referee based on his choice, it is essential to locate a more straight forward strategy for obtaining the global possible reducts.

4.2.4. Integrated reduct from the reducts from clients

The below algorithm provides the method for calculating the global reduct of the information system T=(U,A,C,D) which has n clients say $T_1,T_2,\ldots,T_n$ with the decision on the issues $E_1,E_2,\ldots,E_n$ respectively. Thus, each client itself can be considered as an information system which can be given as $T_i=(U,(A-D)\cup\{E_i), C,\{E_i\}\}$.

4.2.4.1 *Algorithm*

Step 1: Construct discernibility matrix for each client T_i

Step 2: Obtain the Discernibility relation for each T_i

Step 3: Compute reduct of each T_i using Reduct Generation Algorithm

Step 4: Consider a set A=Φ

Step 5: Include all the reducts obtained in step 3 in A and use absorption law in A [for example, if {a,b} and {a,b,c} are in A, then consider the absorption {a,b}∨{a,b,c}={a,b,c}].

Step 6: Write the Integrated reduct.

In the above explained algorithm, it can be viewed clearly that if the reducts from all the clients are known, then the computation of the integrated reduct is straight. The above algorithm is illustrated in the following example:

Example:

The decision table for the first client is given below with C={a,b,c,d, e,f} and D={E_1}.

Table 3.4 Knowledge Representation

	a	B	c	d	E	F	E_1
x_1	1	0	1	1	1	1	1
x_2	0	1	0	1	1	0	1
x_3	1	1	1	0	1	1	0
x_4	0	1	0	0	1	1	0

The discernibility matrix is defined in the below table by

Table 3.5 Discernibility Matrix

	x_1	x_2	x_3
x_2	---		
x_3	b,d	a,c,d,f	
x_4	a,b,c,d	d,f	----

Using the reduct generation algorithm, the reducts obtained are {d} and {b,f}.

The decision table for the second client is given below with C={a,b,c,d,e,f} and D={E_2}.

Table 3.6 Decision Table

	A	B	c	d	E	F	E_2
x_1	1	0	1	1	1	1	0
x_2	0	1	0	1	1	0	1
x_3	1	1	1	0	1	1	1
x_4	0	1	0	0	1	1	0

The discernibility matrix is given by

Table 3.7 Knowledge Representation

	x_1	x_2	x_3
x_2	a,b,c,f		
x_3	b,d	---	
x_4	---	d,f	a,c

Using the reduct generation algorithm, here the reducts obtained are {a,d}, {a,c},{a,b,f} and {b,c,f}.

By using the algorithm 4.2.4.1, the integrated reduct of the server is given by {{d},{b,f}}∨{{a,d}, {a,c},{a,b,f}, {b,c,f}}={{a,d}, {a,c},{a,b,f}, {b,c,f}}. By constructing the discernibility matrix for the server by using all the decisions, this can be verified.

The below is the Integrated decision table of the server with C={a,b,c,d, e,f} and D={E_1}.

Table 3.8 Decision Table

	A	B	C	D	e	F	E_1	E_2
x_1	1	0	1	1	1	1	1	0
x_2	0	1	0	1	1	0	1	1
x_3	1	1	1	0	1	1	0	1
x_4	0	1	0	0	1	1	0	0

Here, the discernibility matrix is given by

Table 3.9 Knowledge Representation

	x_1	x_2	x_3
x_2	a,b,c,f		
x_3	b,d	a,c,d,f	
x_4	a,b,c,d	d,f	a,c

The reducts obtained are {a,d}, {a,c},{a,b,f}, {b,c,f} by using the Reduct generation algorithm and hence it is verified.

From the above example, it is clearly noticed that the Integrated reduct of the server is the OR form of all the client reducts.

In the next section, the process of computing reducts in any decision table of a huge size is discussed. It is necessary to introduce a tool to handle, if the decision table is tremendous in terms of the number of records. Since it will be difficult to form the discernibility matrix and apply reduct generation algorithm. Hence, we introduce an algorithm in the next section to handle huge data, which makes the job easier by partitioning the framework into two subsystems.

4.2.5. Reduct from the Decision Table with huge Data

Consider an information system, which comprises of numerous records. The usual algorithm is to be extended, in order to handle this situation. Consider the information system T=(U,A,C,D) with n records say $x_1, x_2, \ldots, x_n$. Partition T into T_1 and T_2 with respect to records. Let T_1 contains the records $x_1, x_2, \ldots, x_j$ and T_2 contains the records $x_{j+1}, x_{j+2}, \ldots, x_n$. Then the discernibility matrix of T can be defined as Dis(T_1) and Dis(T_2) where Dis(T_1) and Dis(T_2) be the discernibility matrices of T_1 and T_2 respectively, and Between Dis (T_1 , T_2) represents the between discernibility matrix obtained by considering the imparity between the elements of T_1 and T_2 . The reduct which is found in Between Dis (T_1 , T_2) is termed as the ***between reduct*** and is denoted by Bet_Red(T_1 , T_2).

By using the below algorithm, the integrated reduct of this case can be generated.

4.2.5.1. *Algorithm*

Step 1: Construct discernibility matrix for T_i, T_2 and between the matrix of T_i, T_2.

Step 2: Get the Discernibility relation for the three.

Step 3: Compute the reduct of each of the three utilizing Reduct Generation Algorithm and denote them as Red(T_i),Red (T_2) and Bet_Red(T_1 , T_2) respectively.

Step 4: Combine Red(T_1) and Red(T_2) by utilizing all the possible unions between the elements from various sets, say Red($T_1 \cup T_2$) and apply absorption law to it [for illustration, if {a,b} and {a,b,c} are in Red($T_1 \cup T_2$), then consider the absorption {a,b}$\wedge${a,b,c}={a,b}].

Step 5: Combine Red($T_1 \cup T_2$) and Bet_Red(T_1 , T_2) by utilizing all the possible unions between the elements from various sets and apply absorption law to it [for illustration, if {a,b} and {a,b,c} exist, then consider the absorption {a,b}$\wedge${a,b,c}={a,b}].

Step 6: Note the integrated reduct acquired in step 5.

The above algorithm can be represented by the accompanying illustration:

Example:

Consider the discernibility matrix of the records {x_1,x_2,…, x_8} with the attributes {a,b,c,d,e,f}.

Table 4.1 Discernibility Matrix

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_2	a,b						
x_3	b,c,d	a,b					
x_4	---	a,b,d	a,c				
x_5	b,d,e	a,e	a,b,e	a,f			
x_6	a,e,f	a,f	---	---	a,e		
x_7	a,e,f	a,f	a,e	a,e,f	---	a,d,f	
x_8	---	---	a,f	a,e	a,d,e	---	a,d,f

If the system is subdivided into two subsystems namely $T_1=\{x_1,x_2,x_3,x_4\}$ and $T_2=\{x_5,x_6,x_7,x_8\}$, $Red(T_1)=\{\{a,b\},\{b,c\},\{a,c\}\}$; $Red(T_2)= \{\{a\},\{e,f\}\}$and $Bet\_Red(T_1 , T_2) =\{\{a,b\}, \{a,d\}, \{a,e\},\{e,f\}\}$. Hence, $Red(T_1 \cup T_2)=\{\{a,b\},\{a,c\},\{b,c,e,f\}\}$. The integrated reduct is provided by $\{\{a,b\},\{b,c,e,f\},\{a,c,d\},\{a,c,e\}\}$.

By utilizing the iterative process of the same procedure, this algorithm can be developed for huge databases.

Sometimes, in the information system, some of the decision attributes may depend on some other decision attributes. In the next section, the reduct generation algorithm is developed for conditional decision attributes.

4.3. Reduct generation algorithm for conditional decision attributes

Let us consider an information system T=(U,A,C,D) where D is the conditional attribute denoted by E/F where the reduct is to be generated using E under F, say, conditional reduct of E under F.

This algorithm is given as below:

4.3.1 Algorithm

1. Begin
2. Compute the discernibility matrix for T_1=(U,A,C,F)
3. Using reduct generation algorithm, compute reducts, say $X_1,X_2,\ldots,X_t$

    For j=1 to t do

        begin

            Consider the information system T_j=(U,A,X_j,E)

            Using reduct generation algorithm Compute Conditional Reducts

List the reducts

End

4. end

The above algorithm can be portrayed by the following example:

Example:

Consider an information system T=(U,A,C,D) given as follows with C={a,b,c,d} and D={E, F}.

Table 4.2 Information system

	a	b	c	D	E	F
x_1	1	0	2	1	1	1
x_2	1	0	2	0	1	1
x_3	1	2	0	0	2	2
x_4	1	2	2	1	1	0
x_5	2	1	0	0	2	2
x_6	2	1	1	0	0	2
x_7	2	1	2	1	1	1

The discernibility matrix of (U,A,C,{F}) is given by

Table 4.3 Discernibility Matrix

	x_1	x_2	x_3	x_4	x_5	x_6
x_2	---					
x_3	b,c,d	b,c				
x_4	B	b,d	c,d			
x_5	a,b,c,d	a,b,c	---	a,b,c,d		
x_6	a,b,c,d	a,b,c	---	a,b,c,d	---	
x_7	---	---	a,b,c,d	a,b	c,d	c,d

The reducts are computed as {b,c} and {b,d}, by using the reduct generation algorithm.

Now consider the information system T_1 = (U,A,{b,c},{E}}

Table 4.4 Information System

	B	c	E
x_1	0	2	1
x_2	0	2	1
x_3	2	0	2
x_4	2	2	1
x_5	1	0	2
x_6	1	1	0
x_7	1	2	1

For the above Information system, the discernibility matrix is given by

Table 4.5 Discernibility Matrix

	x_1	x_2	x_3	x_4	x_5	x_6
x_2	---					
x_3	b,c	b,c				
x_4	---	---	C			
x_5	b,c	b,c	---	b,c		
x_6	b,c	b,c	b,c	b,c	c	
x_7	---	---	b,c	---	c	C

Hence one of the conditional reducts of E under F is {c}. Similarly, for the reduct {b,d}, the conditional reduct can be computed.

CHAPTER 5

INFORMATION LANGUAGE

This chapter introduces the concept of information language, which is associated with every information system. The information language is used to describe decision rules and decision algorithms in a syntactical way, which allows employing standard logical methods to analyze and investigate these concepts.

5.1 Syntax of the Information Language

With every information system S = (Univ, Att, Val, f) we associate an information language L_S (L-when S is understood), which consists of terms, formulas and decision algorithms.

Terms are built up from some constants by means of Boolean operations +, . , -; we assume that 0,1 are constants and Att, Val is finite sets of constants called attributes and attribute values, respectively.

The set of terms is the least set satisfying the conditions:

a) Constants 0 and 1 are terms in L.
b) Any expression of the form (a: = v), where a ∈ Att where v ∈ $Val_{a)}$ - is a term in L.
c) If s and t are terms in L, so are –t, (t+s) and (t.s) (or simple ts). In what follows we shall drop the unnecessary parenthesis in a usual way in terms. For example the following

-((a:=0) + (b:=2))(c:=1),

(a:=2)(-(b:=1))

are terms in a certain information language.

The set of formulas in an information language L in the least set satisfying the conditions:

a) Constants T (for truth) and F (for falsity) are formulas in L.
b) If t and s are terms in L, then t = s and t,s are formulas in L.
c) If φ *and* Ψ are formulas in L, then (φ ^ Ψ), φ *V* Ψ and (~φ) are formulas in L.

   For example the following

   (a: = 0) = (b: = 2),

   (a: = 1) (b: = 0)   = (c: = 2),

   (a: = 2) -> (b: = 0) + (c: = 1) ,

   ((a: = 0) -> (b: = 0)) U ((a: = 1) -> (c: = 0))

   are formulas in a certain information language

5.2 The meaning of terms and formulas

In this section, we shall define formally the meaning of the terms and formulas in an information system S=(Univ, Att, Val, f). Terms are intended to mean subsets of the universe and the meaning of formulas is the truth of falsity.

In order to define the meaning of terms and formulas, we shall employ the meaning function $g_S : Ter \times For \to P(Univ) \cup \{T, F\}$, where Ter and For denote the set of all terms and formulas, respectively.

The meaning function for terms is defined as follows (we omit the subscript S if S is understood):

a) g(0) = Ø, g(1) = Univ.

b) g(a: = v) = {x ∈ Univ : f(x,a) = v}.

c) g(-t) = Univ – g(t),

g(t+s) = g(t) ∪ g(s),

g(ts) = g(t) ∩ g(s).

For example, the meanings of certain terms in the information system shown in the above table are given below:-

Table 5.1 Information System

Univ	A	B	c
x_1	1	0	2
x_2	0	1	1
x_3	2	0	0
x_4	1	0	2
x_5	1	0	0
x_6	0	1	1
x_7	2	0	0
x_8	1	0	0
x_9	1	0	2
x_{10}	0	1	1

$$g((a: = 1)(b: = 0)) = \{x_1, x_4, x_8, x_8\},$$

$$g((a: = 0) + (c: = 2)) = \{x_1, x_2, x_5, x_7, x_9, x_{10}\},$$

$$g(-a: = 2)) = \{x_1, x_2, x_4, x_5, x_7, x_8, x_9, x_{10}\}.$$

The meanings of formulas are defined in the following way:

a) g(T) = T; g(F) = F.

b) $g(t = s) = \begin{cases} T, if\ g(t) = g(s), \\ F, if\ g(t) \neq g(s) \end{cases}$

c) $g(t \rightarrow s) = \begin{cases} T, if\ g(t) \subseteq g(s), \\ F, otherwise. \end{cases}$

d) $g(\sim\varphi) = \begin{cases} T, if\ g(\varphi) = F, \\ F, if\ g(\varphi) = T. \end{cases}$

e) $g(\varphi \vee \Psi) = g(\varphi) \vee g(\Psi)$

f) $g(\varphi \wedge \Psi) = g(\varphi) \wedge g(\Psi)$

if $g_s(\varphi) = T$, we say that φ is true in S; if $g_s(\varphi) = F$ then φ is said to be false in S. If φ *is true* in S, we shall write $\models_s \varphi$ or simple $\models \varphi$ when S is known.

if $\models_s$ (t = s) we say that t and s are equivalent in s; if $\models_s (t \to s)$ we say that the term t implies the term s in S.

Examples of false and true formulas in the information system given in the above table are shown below:

(a: = 0) = (b: = 1) – true,

(a: = 2) = (c: = 0) – false,

(c: = 0) = (b: = 0) – true,

(c: = 1) = (b: = 0) – false

For the transformation of terms, we shall use the axioms of Boolean algebra and the following specific axiom

$$(a: = v) = -\sum_{\substack{u \neq v \\ u \in val_a}} (a: = u).$$

For the transformation of formulas, we shall employ the axioms of propositional calculus.

5.3 Normal form of terms

A term t in L is A-elementary (A ⊆ Att) if t = $\prod_{a \in A}(a: = v_a)$.

A term t in L is in A-normal from if t = ∑s, where all s are A-elementary. For example the terms

(a: = 1) (b: = 0), (a: = 2) (b: = 1), (a: = 0) (b: = 1)

are {a,b}=elementary and the terms

(a: = 2) (b: = 1) (c: = 2),
(a: = 0) (b: = 0) (c: = 1)

are {a,b,c}-elementary.

The term

(a: =1)(b: = 0) + (a: = 2)(b: = 1)

is in {a,b} – normal form and the term

(a: =2)(b: = 1)(c: = 2) + (a: =0)(b: = 0)(c: = 1)

is in {a,b,c} – normal form.

Let S = (Univ, Att, Val, f) be an information system, A ⊆Att subset of attributes, and L_A - an information language with the set of attributes A.

5.3.1 Property

For every term t in L_A there exists the terms s in L_A, in an A-normal form such that $\models_s t = s$; s is referred to as the A-normal form of t in L_A.

For example the {a,b} – normal form of the term (a: = 1) is the term (a: = 1)(b: = 0) + (a: = 1)(b: = 1).

Subset X ⊆ Univ is said to be A-discernible in L(A ⊆ Att) if there exists a term t in L such that $g_s(t) = X$; the term t is called A-description of X in L.

If set X ⊆ Univ is non A-describable in L, then there exist terms t and s such that $g_s(t) = \underline{A}X$ and $g_s(s) = \overline{A}$ X, called $\underline{A}$ - lower and $\overline{A}$ - upper description of X in L, respectively. This is to mean that some subset of objects is described by a given subset of attributes not exactly but with some approximation only.

For example, the subset X = $\{x_3, x_4, x_7\}$ of objects in the information system shown in the above table has the following A-lower and A-upper description (A={a,b,c}).

X's A-lower description is

(a: = 2)(b: = 0)(c: = 0);

X's A-upper description is

(a: = 2)(b: = 0)(c: =0) + (a: = 1)(b: = 0)(c: = 2).

5.4 Decision Rules

Any formula of the form t-> s is called a decision rule in L; t is referred to as a condition and s – as a decision of the decision rule, respectively. If the decision rules t->s is true we shall also say that it is consistent; otherwise, the decision rule is inconsistent.

Let t->s be a decision rule in L and let A, B ⊆ Att be sets of l attributes which occur in t and s respectively, we shall call t->s (A, B) – decision rule. If

A and B are one element sets, for the sake of simplicity, we shall use the expression (a,b) – decision rule.

Let S = (Univ, Att, Val, f) be an information system and t-> s (A, B) – decision rule in S.

We say that the (A,B) – decision rule is deterministic in S, if $g_s(s) \in B^*$, i.e. $g_s(su)$ in a description of a certain equivalence class of the equivalence relation B; otherwise the decision rule is non-deterministic.

We say that a(A,B) – decision rule t->s is in (A,B) – normal form if t and s are in (A,B) – normal form.

5.4.1 Property

A (A, B) – decision rule t -> s is consistent in S iff all non-empty elementary terms occurring in (A,B) – normal form of t occur also in the (A,B) – normal form of s.

This property enables us to prove the validity of any decision rule in a simple syntactical way.

Example

Consider the information system shown in the table below

Univ	A	B
x_1	1	0
x_2	1	1
x_3	0	2

Let us check whether the decision rule (a: = 1) -> (b: = 0) is consistent or not.

The (a,b) – normal form of the rule is

(a: = 1)(b: =0) + (a: = 1)(b: = 1) -> (a: = 1)(b: = 0)

Because the elementary term (a: = 1)(b: = 1) occurs only in the condition of (a,b) – normal form of the rule, for the rule (a: = 1) - > (b: = 0) is inconsistent.

We can also check whether the decision rule inconsistent or not, using the meaning function (semantics), namely:

$g_s(a := 1) = \{x_1, x_2\}$

and

$$g_s(b := 0) = \{x_1\},$$

hence $g_s(a := 1) \subseteq g_s(b := 0)$ and the decision rule is inconsistent.

On the other hand the decision rule (b: =0) -> (a: = 1) is consistent because (a,b) – normal form of the rule is in the form:

(a: = 1)(b: = 0) -> (a: = 1) (b: = 0) + (a: = 1) (b: = 1)

and the only (a,b) – elementary term (a: = 1) (b: = 0) in the condition of the decision rule occurs also in the decision of the rule.

Employing the definition of semantics, we have

$g_s(a := 1) = \{x_1, x_2\}$ and $g_s(b := 0) = \{x_1\}$, hence $g_s(b := 0) \subset g_s(a := 1)$ and the decision rule is consistent.

Let us notice that both rules are deterministic in the information system.

5.5 Decision Algorithms

Any finite set of decision rules in L is called a decision algorithm in L. An example of a decision algorithm is shown below:

(a: = 1) -> (b: = 2)(c: = 1),

(b: = 2) + (a: = 1) -> (c: = 2),

(a: = 0) + (b: =2) -> (c: = 1).

With every decision algorithm a = $\{t_1 \rightarrow s_i\}_n$, 1 ≤ i ≤ m, in L we associate the formula

$$\overline{\psi}\, a = \bigwedge_{i=1}^{n} \{t_1 \rightarrow s_i\}$$

Called the decision formula of a in L.

A decision algorithm is said to be consistent if all its decision rules are consistent, otherwise, the decision algorithm is inconsistent.

5.5.1 Property

A decision algorithm a in L is consistent (in S) iff $\models_s$ Ψa.

Example

It is easy to see that the decision algorithm

(a: = 1)(b: =0) -> (c: = 2) + (c: = 0),

(a: = 0) -> (c: = 1),

(a: = 2) + (b: = 1) -> (c: = 0) + (c: = 1).

The decision algorithm

(a: = 1) -> (c: =2)

(a: = 0) -> (c: = 0).

is consistent in the system.

A decision algorithm is deterministic in S if all its decision rules are deterministic in S; otherwise, the algorithm is non-deterministic. For example, the two above decision algorithms are non-deterministic. If A and B are the sets of all attributes occurring in the conditions and decisions of the decision rules of the algorithm a, then a will be called the (A, B) – decision algorithm and denoted as a(A, B).

5.5.2 Property

(A, B) – a decision algorithm is deterministic in S iff A → B in S.

A (A, B) – decision algorithm is total in S if for every equivalence class X of the equivalence relation B there exists a decision rule t -> s in A such that $g_s(s) = X$; otherwise the decision algorithm is partial in S.

Example

The algorithm

(a: = 1)(b: = 0) -> (c: = 2),

(a: = 0) -> (c: = 1),

(a: = 2) = (b: = 1) -> (c: = 0)

The decision algorithm

(a: = 1)(b: = 0) -> (c: = 2),

(a: = 0)(b: = 1) -> (c: = 1)

is partial in the information system.

The following property can be used as a transformation rule for decision algorithms:

5.5.3 Property

$$\vDash_s \bigwedge_{i=i}^{n} (t_i \rightarrow s) \equiv (\sum_{i=1}^{n} t_i \rightarrow s) \qquad \textbf{(5.1)}$$

where ≡ is defined in the usual way.

The following property establishes the relationship between the consistency and determinism of a decision algorithm.

5.5.4 Property

If A → B in S, then $\vDash_s$ Ψa (A,B).

5.6 Examples

In this section, we will depict previously introduced notions by means of three examples: discrimination analysis, learning from examples and decision tables.

5.6.1 Discrimination Analysis

Suppose we are given a data file, for example, concerning patients suffering from a certain disease. With every patient, several items of information (symptoms) are associated. Besides the patients are classified according to a certain pre-assumed rule, for example, age, disease advance, etc. The classification can be based on existing symptoms or it can be given by an expert. The main problem of discrimination analysis consists in describing each class of the classification in terms of available symptoms.

Thus the problem can be reduced to that discussed in previous sections, namely, we can treat the data file as an information system and ask whether $A \rightarrow c$ or not, where c is an attribute representing the classification and A is the set of attributes representing symptoms. The second problem is to find reducts of A in order to find the minimal number of symptoms necessary to classify objects properly.

An example given below explains the idea of discrimination more exactly.

Example

Let us consider an information system shown in the table

Univ	a	B	C	D
x_1	1	0	2	1
x_2	0	1	1	2
x_3	2	0	0	3
x_4	1	1	0	1
x_5	1	0	2	2
x_6	2	0	0	3
x_7	0	1	1	2
x_8	0	1	1	2

Let attributes a,b,c represent some “symptoms” and the attribute d, the classification of objects.

Our first problem consists in checking whether {a,b,c} → d, and the second one is the reduction of "symptoms" {a,b,c}.

Let us denote {a,b,c} = A, let us compute A^* and d^*. The A-elementary sets are classes of the classification A^* and are as follows:

$$X_1 = \{x_1, x_5\},\ X_2 = \{x_2, x_7, x_8\},$$
$$X_3 = \{x_3, x_6\},\ X_4 = \{x_4\}$$

Classes of the classification d^* are:

$$Y_1 = \{x_1, x_4\},$$
$$Y_2 = \{x_2, x_5, x_7, x_8\},$$
$$Y_3 = \{x_3, x_6\}.$$

The corresponding approximations are as follows:

$$\underline{A}\, Y_1 = X_4 = \{x_4\}, \qquad \underline{A}\, Y_1 = X_7 \cup X_4 = \{x_1, x_4, x_5\},$$
$$\mathrm{B}\eta_a\,(Y^*) = X_1 = \{x_1, x_5\}, \qquad \gamma_A(Y_1^*) = 1/3,$$
$$\underline{A}\, Y_2 = X_2 = \{x_4, x_7, x_8\}, \quad \underline{A}\, Y_2 = X_2 \cup X_2 = \{x_1, x_2, x_4, x_5 x_7, x_8\},$$
$$\mathrm{B}\eta_A(Y_1^*) = X_3 = \{x_1, x_4\}, \qquad \gamma_A(\,Y_2) = ½,$$
$$\underline{A}\, Y_3 = X_3 = \{x_3, x_6\}, \quad \overline{A}\, Y_3 = X_3 = \{x_3, x_6\},$$
$$\mathrm{B}\eta_A(Y_3^*) = \varnothing\ , \quad \gamma_A(Y_3^*) = 1,$$
$$Int_A(d^*) = \underline{A}\, Y_1 \cup \underline{A}\, Y_2 \cup \underline{A}\, Y_3 = X_2 \cup X_3 \cup X_4 = \{x_2, x_3, x_4, x_6 x_7, x_8\},$$
$$\gamma_A(d^*) = \frac{card\ Int_A(d^*)}{card\ Univ} = 6/8 = ¾.$$

That is to mean that A <u>0</u>.75 d.

Thus we are unable to classify objects according to the classification d^* using the set of attributes A. Only 6 out of 8 objects can be classified correctly using the set attributes A (namely objects$\{x_2, x_3, x_4, x_6 x_7, x_8\}$).

The accuracies of particular classes are shown below:

Class	a
1	0,5
2	0,5
3	1,0

That means classes 1 and 2 are roughly discernible by the set of attributes A and class 3 is discernible by the set of attributes.

It is easy to compute that the core of A with respect to d is the empty set and the set A has three reducts { a,b}, {b,c}, {a,c} with respect to d. Thus, any pair of attributes from A will suffice to classify objects with the same accuracy as that provided by the whole set of attributes of A.

We can also give a decision algorithm, which enables us to classify objects into proper classes using properties expressed by attributes from A.

An example of a decision algorithm is shown below:

(a:=1)(b:-=1) -> (d:=1)

(a=1)(b:=0) -> (d:=1)(d:=2)

(a:=0) -> (d:=2)

(a:=2) -> (d:=3)

Let us notice that the algorithm is non-deterministic, total and consistent. This scheme of data analysis can be employed in psychology, sociology, agriculture, engineering, and other areas.

5.6.2 Learning from examples

Suppose we are given a finite set of Univ of objects. Elements of Univ are called training examples and univ is called a training set. Assume further that Univ is classified into disjoint classes $X_1, X_2, \ldots, X_n\ (n > 2)$by a teacher (expert, environment etc). The classification represents the teacher knowledge of objects from Univ in terms of attributes from preassumed set A. Descriptions of objects in terms of attributes from A represent the student knowledge of the objects from Univ.

We can say that the student has a syntactical knowledge and the teacher- the semantical knowledge – about objects from Univ.

The problem of learning from the example is, whether the student knowledge can be matched with the teacher's knowledge, or more precisely, whether the teacher's classification can be described in terms of attributes available to the student.

Thus, learning from the example consisting of described classes $X_1, X_2, \ldots, X_n$ in terms of attributes from A, or more exactly, in finding a decision algorithm which provides the teacher's classification on the basis of the properties of an object expressed in terms of attributes from A.

It is easily seen that the problem of learning from examples can be formulated in terms of notations introduced before. Training examples from the universe Univ, "Students" attributes A and teacher attribute e-is the set of attributes Att.

Thus the problem of learning from examples is reduced to the question whether A -> e (or whether e^* is A-discernible),i.e.– whether there exists an algorithm to ”learn” classification e^* by attributes of training examples.

This can be easily done by using methods discussed in the previous sections. For example, let us consider the following information system:

Example

Univ	A	b	C
x_1	1	0	2
x_2	0	1	1
x_3	2	0	0
x_4	1	0	2
x_5	1	0	0
x_6	0	1	1
x_7	2	0	0
x_8	1	0	0
x_9	0	1	1
x_{10}	2	0	0
x_{11}	1	0	0
x_{12}	1	0	2

where Univ is the training set, A = {a,b} are students’ attributes and c is teacher attribute.

In order to check whether the concepts represented by examples $x_1, \ldots . x_{12}$ can be learned using attributes from A-according to the teacher knowledge expressed by the classification c^*, we have to compute degree k of dependency A $\underline{k}$ c.

In order to do so let us compute classes of the classification c^*,which are as follows:

$$Y_1 = \{x_1, x_4, x_{12}\},$$
$$Y_2 = \{x_2, x_6, x_9\}.$$
$$Y_3 = \{x_3, x_5, x_7, x_9, x_{10}, x_{11}\}$$

And classes of the classification A^*, are given below:

$$X_1 = \{x_1, x_4, x_5, x_8, x_{11}, x_{12}\},$$
$$X_2 = \{x_2, x_6, x_9\},$$
$$X_3 = \{x_3, x_7, x_{10}\}$$

The corresponding approximations are:

$$\underline{A}\, Y_1 = \emptyset, \quad \overline{A}\, Y_1 = X_1,$$
$$\text{B}\, \eta_A(Y_1^*) = X_1, \quad \gamma_A(Y_1^*) = 0,$$
$$\underline{A}\, Y_2 = X_2, \quad \overline{A}\, Y_2 = X_2,$$
$$\text{B}\, \eta_A(Y_2^*) = \emptyset, \quad \gamma_A(Y_2) = 1,$$
$$\underline{A}\, Y_3 = X_3, \quad \overline{A}\, Y_3 = X_1 \cup X_3,$$
$$\text{B}\, \eta_A(Y_3^*) = X_1, \quad \gamma_A(Y_3) = 0$$
$$Int_A(c^*) = \underline{A}\, Y_1 \cup \underline{A}\, Y_2 \cup \underline{A}\, Y_3 = X_2 \cup X_3 = \{x_2, x_3, x_6, x_7, x_9, x_{10}\},$$
$$\gamma_A(c^*) = \frac{6}{12} = 0.5$$

Thus the class Y_1 - is internally A-indiscernible, $Y_2 -$ is A-discernible and Y_3 *is* roughly A-discernible.

Thus, it is impossible to learn positive instances of Y_1,but is possible to learn negative instances of Y_1, $(if\ x \cup X_2 \cup X_3$ we know that x is not in Y_1).

In other words, it is impossible to classify correctly $\{x_1, x_4, x_{12}\}$ by checking their features expressed by attributes a and b.

The set Y_2 can be learned, i.e. all elements of Y_2 can be classical property examining their features a and b.

Set Y_3 can be learned roughly, i.e. only examples $\{x_3, x_7, x_{10}\}$ can be recognized on the basis of a and b as elements of Y_3; objects $\{x_2, x_6, x_9\}$ can be excluded being members of Y_1 and $X_1 = \{x_1, x_4, x_5, x_8, x_{11}, x_{12}\}$ is the boundary of Y_3, i.e. if cannot be decided whether elements of X_1 belong to Y_3 or not – employing the attributes a and b.

The corresponding decision algorithm is shown below:

(a: = 1)(b: = 0)(c: = 2) + (c: = 0),

(a: = 0)(c: = 1),

(a: = 2)(c: = 0).

The algorithm is non-deterministic, total and consistent.

5.7 Decision Tables

Decision Tables are the important tools in computer applications. The application of rough sets to decision table analysis is shown in this section. The proposed approach seems to be very suitable to formulate and solve many basic problems concerning decision table analysis and implementation.

Decision table can be considered as an information system in which attributes are divided into two classes called condition and decision attribute, denoted as Con and Dec respectively.

Each row of the decision table is called a decision rule. In other words, the decision rule in a function f_x: Att → Val such that $f_x(a) = f(x, a)$ for every x ∈ Univ and a ∈ Att. The restriction of f_x to condition attributes, denoted f_x/Con, will be called condition and the restriction of f_x to Dec is denoted f_x/Dec - the decision of the rule f_x. Decision rules will be written in the form p -> g, where p is the condition and g is the decision of the rule. An example of the decision table is shown below:

Example

Let us notice that the decision rule (a: = 0)(b: = 1)(c: = 1) -> (d: = 1)(c: = 2) is non-deterministic whereas the rule (a: = 2)(b: = 0)(c: = 0) -> (d: = 1)(e: = 1) is deterministic in the table below.

Univ	a	B	c	d	E
1	1	0	2	2	0
2	0	1	1	1	2
3	2	0	0	1	1
4	1	1	0	2	2
5	1	0	2	0	1
6	2	2	0	1	1
7	2	1	1	1	2
8	0	1	1	0	1

A decision table is deterministic if all its decision rules are deterministic, otherwise, the decision table is non-deterministic.

5.7.1 Property

A decision table S is deterministic iff Con → Dec in S.

A decision table S is said to be roughly deterministic if Con k Dec and 0 < k < 1; a decision table is totally non-deterministic if Con 0 Dec in S.

Example

The decision table shown above is non-deterministic and we have Con .5 Dec, and the table can be decomposed into two decision tables as shown below:

Table 5.2 Non-Deterministic Decision Table 1

Univ	a	b	c	d	E
3	2	0	0	1	1
4	1	1	0	2	2
6	2	2	0	1	1
7	2	1	1	1	2

Table 5.3 Non-Deterministic Decision Table 2

Univ	a	b	c	d	E
1	1	0	2	1	0
2	0	1	1	1	2
5	1	0	2	0	1
8	0	1	1	0	1

The deterministic decision table is shown in Tab. 5.2 and the non-deterministic table is shown in Tab. 5.3.

Let us notice that $\gamma_{Con}(Dec^{*}) = 1/2$ in the table. It can easily be shown that the set of condition attributes in decision table are shown in Tab. A has only one reduct {a,b} and the set of decision attributes is independent.

Thus the decision table can be presented as shown in Tab. 5.3

Table 5.4 Non-Deterministic Decision Table 3

Univ	*A*	*B*	*d*	*E*
1	1	0	2	0
2	0	1	1	2
3	2	0	1	1
4	1	1	2	2
5	1	0	0	1
6	2	2	1	0
7	2	1	1	2
8	0	1	0	1

The next example shows a more complicated simplification of a decision table.

Example

The decision table shown below is deterministic.

Table 5.5 Deterministic Decision Table

Univ	*a*	*B*	*C*	*d*	*e*	*F*
1	3	2	2	2	2	4
2	3	2	2	1	2	4
3	2	2	2	1	1	4
4	2	2	2	2	1	4
5	3	2	2	3	2	3
6	3	3	2	3	2	3
7	4	3	2	3	2	3
8	4	3	3	3	2	2

9	4	4	3	3	2	2
10	4	4	3	2	2	2
11	4	3	3	2	2	2
12	4	2	3	2	2	2
13	3	3	2	2	2	4

Attributes a,b,c,d are conditional attributes and e,f are decision attributes in the table.

The set {a,b,c,d} of condition attributes are independent.

Now we can compute the reducts of Con with respect to the classification$\{e, f\}^*$, and we get the following reducts:

{a,b} for e: = 2 and f: = 4,

{a} for e: = 1 and f: = 4,

{e,a} for e: = 2 and f: = 3,

{e} for e: = 2 and f: = 2.

Hence we can simplify the table as shown below:

Table 5.6 Simplified Decision Table

Univ	a	B	C	D	e	f
1`	3	-	-	1	2	3
2`	3	-	-	2	2	4
3`	2	-	-	-	1	4
4`	-	-	2	3	2	3
5`	-	-	3	-	2	2

Consequently, we get the following decision algorithm:

1`: (a: = 3)((d: = 1) + (d: = 2)) -> (e: = 2)(f: = 4),

2`: (a: = 2) -> (e: = 1)(f: = 4),

3`: (c: = 2)(d: = 3) -> (e: = 2)(f: = 3),

4`: (c: = 3) -> (e: = 2)(f: = 2).

CHAPTER 6

ROUGH SET: ANALYSIS OF FUZZY SETS USING THRESHOLDS

In an expert system with the crisp knowledge base, if the input is fuzzy natured, it is necessary to approximate it to a crisp set [20] so that it can be analyzed using the available knowledge.

Consider a knowledge base about objects classified based on their color. Given an input object whose color is a combination of two or more classes of the color of different ratios. Determining to which class this input object belongs is difficult.

Consider another case where the knowledge is based on voice recognition; i.e., similar voices are treated as similar objects, when a voice signal is received it is to be processed. So, it is necessary to introduce a tool to resolve such problem.

This chapter introduces the lower and upper rough approximation of a given fuzzy set under the given threshold α, which gives a strong α cut of the given fuzzy set. Further, it introduces the significance of it in defuzzification problems and the approximations of fuzzy predicates.

Sometimes, in decision-making, it is necessary to take proper decisions for the elements whose fuzzy membership values lie in some interval. In those cases, analyzing with a single threshold is time-consuming and ineffective. Here, it would be more effective to introduce two thresholds [16]. It can be noted that by making the higher threshold to 1, one can arrive at the definition with one threshold.

Here, the concept of one threshold on fuzzy sets is extended to any fuzzy set with two thresholds. In each case, the resultant rough set is an ordered pair of two crisp sets. Additionally, their properties are discussed.

Now, it is necessary to discuss the possible values of thresholds taken in (0,1).

6.1. Analysis of fuzzy set using a threshold

First, the set D is constructed as follows, which is used as the domain of thresholds.

Consider a set D, which is termed as **R-domain** [14], satisfying the below properties:

a) $D \subset (0,1)$
b) Ignore the values $\mu_A(x)$ and $\mu_A^c(x)$ $\forall x \in U$ from the domain D, if a fuzzy set A is under computation, if they exist.
c) The values which are removed in the step (b) may be included in D provided A must not involve in further computation, after the computation using A.

Consider $U=\{x_1,x_2,\ldots,x_n\}$ as the universe of discourse. Let $\alpha,\alpha_1,\alpha_2,\beta$ be the thresholds. D is developed utilizing the fuzzy sets A and B, assume one of the values from the domain D.

6.1.1. Property:

$A[\alpha_1] \cup A[\alpha_2] = A[\alpha]$ where $\alpha = \min(\alpha_1,\alpha_2)$

Proof:

$x_i \in A[\alpha_1] \cup A[\alpha_2] \Leftrightarrow x_i \in A[\alpha_1]$ or $x_i \in A[\alpha_2] \Leftrightarrow \mu_A(x_i) > \alpha_1$ or $\mu_A(x_i) > \alpha_2$ $\Leftrightarrow \mu_A(x_i) > \alpha \Leftrightarrow x_i \in A[\alpha]$

6.1.2. Property:

$A[\alpha_1] \cap A[\alpha_2] = A[\alpha]$ where $\alpha = \max(\alpha_1, \alpha_2)$

Proof:

Similar to 6.1.1

6.1.3. Property:

$(A \cup B)[\alpha] = A[\alpha] \cup B[\alpha]$

Proof:

$x_i \in (A \cup B)[\alpha] \Leftrightarrow \mu_{A \cup B}(x_i) > \alpha \Leftrightarrow \max(\mu_A(x_i), \mu_B(x_i)) > \alpha \Leftrightarrow \mu_A(x_i) > \alpha$ or $\mu_B(x_i) > \alpha \Leftrightarrow x_i \in A[\alpha]$ or $x_i \in B[\alpha] \Leftrightarrow x_i \in A[\alpha] \cup B[\alpha]$

6.1.4. Property:

$(A \cap B)[\alpha] = A[\alpha] \cap B[\alpha]$

Proof:

Similar to 6.1.3

6.1.5. Property:

$A^c[\alpha] = A[1-\alpha]^c$

Proof:

$x_i \in A^c[\alpha] \Leftrightarrow \mu_A{}^c(x_i) > \alpha \Leftrightarrow 1-\mu_A(x_i) > \alpha \Leftrightarrow \mu_A(x_i) < 1-\alpha \Leftrightarrow x_i \notin A[1-\alpha] \Leftrightarrow$ $x_i \in A[1-\alpha]^c$

***6.1.6. Property**:*

$(A \cup B)^c[\alpha] = A^c[\alpha] \cap B^c[\alpha]$

***Proof**:*

$(A \cup B)^c[\alpha] = (A \cup B)[1-\alpha]^c$ [by 6.1.5]

$= A[1-\alpha]^c \cup B[1-\alpha]^c$ [by 6.1..3]

$= A^c[\alpha] \cap B^c[\alpha]$

***6.1.7. Property**:*

$(A \cap B)^c[\alpha] = A^c[\alpha] \cup B^c[\alpha]$

***Proof**:*

Similar to 6.1.6

The following example illustrates the properties defined above.

***Example**:*

Consider U={a,b,c,d,e,f} as the universe of discourse. Let A=(0.2, 0.4, 0.3,0.5,0.7,0) and B=(0.6,0.8,1,0.4,0.6,0.6). Then D=(0,1)-{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8}. Let $\alpha \in D$, say, $\alpha=0.45$. Then $A[\alpha]$={d,e}, $B[\alpha]$={a,b,c,e,f}, $A^c[\alpha]$= {a,b,c,d,f}, $B^c[\alpha]$={d},$(A \cup B)[\alpha]$=U,$(A \cap B)[\alpha]$= {e}, $(A \cup B)^c[\alpha]$= {d}and $(A \cap B)^c[\alpha]$ ={a,b,c,d,f}. Also $A[1-\alpha]^c$={a,b,c,d,f}. Here, it can be observed that properties 6.1.3 to 6.1.7 hold.

Let $\alpha_1=0.45$ and $\alpha_2=0.55$. Then $A[\alpha_1]$={d,e}, $B[\alpha_1]$ and $A[\alpha_2]$={e}. Hence $A[\alpha_1] \cup A[\alpha_2] = A[\alpha_1]$ and $A[\alpha_1] \cap A[\alpha_2] = A[\alpha_2]$. Thus, the properties 6.1.1 and 6.1.2 are verified.

Using the mathematical tool derived in section 6.1, we introduced rough set approach on fuzzy sets using a threshold in [14].

6.2. Rough set approach on fuzzy sets using α

Consider Ψ be any partition of U, say $\{B_1,B_2,\ldots,B_t\}$. The lower and upper approximations with corresponding to α can be stated as $A_\alpha=\underline{(A[\alpha])}$ and $A^\alpha=\overline{(A[\alpha])}$ respectively for the provided fuzzy set A.

6.2.1 Propositions

Here, the following propositions can be obtained, by utilizing the properties of the rough sets.

6.2.1.1. Proposition:

$(A\cup B)^\alpha=A^\alpha\cup B^\alpha$

Proof:

$(A\cup B)^\alpha=\overline{(A\cup B)[\alpha]}=\overline{A[\alpha]\cup B[\alpha]}=\overline{(A[\alpha])}\cup\overline{(B[\alpha])}=A^\alpha\cup B^\alpha$

6.2.1.2. Proposition:

$(A\cap B)_\alpha=A_\alpha\cap B_\alpha$

Proof:

Similar to 6.2.1.1

6.2.1.3. Proposition:

$(A\cup B)_\alpha\supseteq A_\alpha\cup B_\alpha$

Proof:

Similar to 6.2.1.1

6.2.1.4. Proposition:

$(A\cap B)^{\alpha}\subseteq A^{\alpha}\cap B^{\alpha}$

Proof:

Similar to 6.2.1.1

6.2.1.5. Proposition:

$(A^{c})^{\alpha}=(A_{1-\alpha})^{c}$

Proof:

$(A^{c})^{\alpha}=\overline{(A^{c}[\alpha])}=\overline{(A[1-\alpha]^{c})}=[\underline{(A[1-\alpha])}]^{c}=(A_{1-\alpha})^{c}$

6.2.1.6. Proposition:

$(A^{c})_{\alpha}=(A^{1-\alpha})^{c}$

Proof:

Similar to 6.2.1.5

Example:

Consider U={a,b,c,d,e,f} as the universe of discourse with the partition X={{a,b}, {c}, {d,f}, {e}}. Let A=(0.2,0.4,0.3,0.5,0.7,0) and B=(0.6,0.8,1, 0.4, 0.6, 0.6). Then D=(0,1)-{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8}. Let $\alpha\in D$, say, $\alpha=0.45$.

Then $A[\alpha]$={d,e} and $B[\alpha]$={a,b,c,e,f}. Hence A_{α}={e}; A^{α}={d,e,f}; B_{α}={a,b,c,e} and B^{α}=U.

$A\cup B$=(0.6, 0.8, 1, 0.5, 0.7, 0.6); $(A\cup B)[\alpha]=U$. Hence $(A\cup B)_\alpha=(A\cup B)^\alpha=U$. But, $A_\alpha\cup B_\alpha$ = {a, b, c, e} and $A^\alpha\cup B^\alpha=U$. Hence, $(A\cup B)^\alpha=A^\alpha\cup B^\alpha$ and $(A\cup B)_\alpha\supseteq A_\alpha\cup B_\alpha$.

$A\cap B$=(0.2, 0.4, 0.3, 0.4, 0.6, 0); $(A\cap B)[\alpha]=\{e\}$. Hence $(A\cap B)_\alpha=(A\cap B)^\alpha=\{e\}$. But, $A_\alpha\cap B_\alpha$ = {e} and $A^\alpha\cap B^\alpha=\{d,e,f\}$.Hence, $(A\cap B)_\alpha=A_\alpha\cap B_\alpha$ and $(A\cap B)^\alpha\subseteq A^\alpha\cap B^\alpha$.

A^c=(0.8,0.6,0.7,0.5,0.3,1). $A^c[\alpha]=A^c[0.45]=\{a,b,c,d,f\}$. Hence $(A^c)_\alpha=(A^c)^\alpha=\{a,b,c,d,f\}$. Now, $A[1-\alpha]=\{e\}$. Hence, $A_{1-\alpha}=A^{1-\alpha}=\{e\}\Rightarrow(A_{1-\alpha})^c=(A^{1-\alpha})^c=\{a,b,c,d,f\}$. Therefore, $(A^c)^\alpha=(A_{1-\alpha})^c$ and $(A^c)_\alpha=(A^{1-\alpha})^c$

This work discusses the relation between rough fuzzy sets and rough sets on fuzzy Sets [5, 17] using thresholds.

However, in the expert system works under fuzzy environment, to make decisions, the rough fuzzy sets are to be de-fuzzified by any of the standard methods of defuzzification.

In parallel, in the above situation, by using the threshold by taking the strong α-cut on the fuzzy input, it can be de-fuzzified and on it, the rough approximations can be taken for further decision-making.

Both methods can find several applications in decision-making. The first method is useful in any hybrid system, which deals with rough sets and fuzzy logic. The second method works in a hybrid system, which involves in the hybridization of fuzzy logic, rough sets, and artificial neural networks.

In 1988, Nakarmura [26] gave an introduction to fuzzy rough sets. Using it, in 1989, Dubois and Prade [11] hybridized rough and fuzzy concepts and contributed rough fuzzy and fuzzy rough concepts, which found wide applications in decision-making in a fuzzy environment, which are same as that of rough sets in the crisp environment. In this chapter, the concept of rough fuzzy sets is extended into generalized rough fuzzy sets and theory of rough fuzzy groups is introduced.

6.3 Rough Fuzzy Sets

Let U={x_1,x_2,…,x_n} be the universe of discourse. Let Ψ ={B_1,B_2,….,B_t} be the partition of U, then for any fuzzy subset F of U with μ_F as membership function, the lower and upper rough approximations of F are defined as

$$\mu_{\underline{F}}(B_i) = \inf_{x_j}\{\mu_F(x_j)/x_j \in B_i\} \text{ and} \quad \textbf{(6.1)}$$

$$\mu_{\overline{F}}(B_i) = \sup_{x_j}\{\mu_F(x_j)/x_j \in B_i\} \text{ respectively.} \quad \textbf{(6.2)}$$

The lower approximation of F is given by $\underline{F}$ =($\mu_{\underline{F}}$(B_1), $\mu_{\underline{F}}$(B_2),…., $\mu_{\underline{F}}$(B_t)) and the upper approximation is given by $\overline{F}$ =($\mu_{\overline{F}}$(B_1), $\mu_{\overline{F}}$(B_2),…. $\mu_{\overline{F}}$(B_t))

Here ($\underline{F}$, $\overline{F}$) is called a *rough fuzzy set.*

Example:

Consider U={a,b,c,d,e,f} as the universe of discourse with the partition Ψ ={B_1, B_2, B_3} where B_1={a,c,e},B_2={b,f} and B_3={d}. Consider the fuzzy subset F=(0.2,0.4,0.3,0.6,0.2,0.7) of U. Then $\mu_{\underline{F}}$(B_1)=min {0.2,0.3,0.2}=0.2;

$\mu_{\underline{F}}$(B$_2$)=min {0.4, 0.7}=0.4 and $\mu_{\underline{F}}$(B$_3$)=min {0.6}=0.6, which gives $\underline{F}$=(0.2,0.4,0.6). Similarly, $\mu_{\overline{F}}$(B$_1$) =max {0.2,0.3,0.2}=0.3; $\mu_{\overline{F}}$(B$_2$)=max {0.4, 0.7}=0.7 and $\mu_{\overline{F}}$(B$_3$)= max {0.6}=0.6, which gives $\overline{F}$=(0.3,0.7,0.6).

From the definition of rough fuzzy sets, some of the properties of rough fuzzy sets can be observed [6], which are listed below.

6.3.1 Properties:

a) $\underline{F} \subseteq \overline{F}$

b) $\overline{F \cup G} = \overline{F} \cup \overline{G}$

c) $\overline{F \cap G} \subseteq \overline{F} \cap \overline{G}$

d) $\underline{F} \cup \underline{G} \subseteq \underline{F \cup G}$

e) $\underline{F} \cap \underline{G} = \underline{F \cap G}$

f) $[\overline{F}]^c = \underline{F^c}$

g) $[\underline{F}]^c = \overline{F^c}$

These properties are illustrated by the following example.

Example:

Consider U={a,b,c,d} as the universe of discourse with B$_1$={a,b} and B$_2$={c,d}. Let F=(0.2,0.4,0.4,0.5) and G=(0.3,0.1,0.3,0.2) be the fuzzy subsets of U. Then $\underline{F}$=(0.2,0.4); $\overline{F}$=(0.4,0.5); $\underline{G}$=(0.1,0.2) and $\overline{G}$=(0.3,0.3). Hence, F∪G=(0.3,0.4,0.4,0.5) and F∩G= (0.2,0.1,0.3,0.2) which gives $\underline{F \cup G}$=(0.3,0.4); $\overline{F \cup G}$=(0.4,0.5); $\underline{F \cap G}$=(0.1,0.2) and $\overline{F \cap G}$=(0.2,0.3). From these, the properties 6.3.2 (a) to (e) can be verified. Moreover,

F^c=(0.8,0.6,0.6,0.5) gives $\underline{F^C}$=(0.6,0.5) and $\overline{F^C}$=(0.8,0.6). Now, $[\overline{F}]^c$=(0.6,0.5) and $[\underline{F}]^c$=(0.8,0.6), which shows 6.3.2 (f) and (g).

6.4 Thresholds on Rough Fuzzy Sets

In rough fuzzy sets, for defuzzification of approximations, the threshold α is to be taken from the domain D say Rough Fuzzy Domain or RF-Domain, which can be defined as follows:

a) $D \subseteq (0,1)$
b) If a fuzzy set A is under computation, eliminate the minimum and maximum membership values and the complements of the granules.

While taking the strong α cut of any approximation where $\alpha \in D$, the crisp set can be obtained by taking the union of the elements of the partition which have the membership value more than α.

By using the thresholds from **RF domain**, the following properties can be observed.

6.4.1. Properties:

Consider the rough fuzzy sets $(\underline{A}, \overline{A})$ and $(\underline{B}, \overline{B})$ of the fuzzy sets A and B respectively. Let α be any threshold from the RF domain. Then

a) $\overline{A \cup B}[\alpha] = \overline{A}[\alpha] \cup \overline{B}[\alpha]$

b) $\underline{A \cap B}[\alpha] = \underline{A}[\alpha] \cap \underline{B}[\alpha]$

c) $\underline{A \cup B}[\alpha] \supseteq \underline{A}[\alpha] \cup \underline{B}[\alpha]$

d) $\overline{A \cap B}[\alpha] \subseteq \overline{A}[\alpha] \cap \underline{B}[\alpha]$

e) $\overline{A^C}[\alpha] = \left(\underline{A}[1-\alpha]\right)^C$

f) $\underline{A^C}[\alpha] = \left(\overline{A}[1-\alpha]\right)^C$

These properties can be verified by the following example.

Example:

Consider U={a,b,c,d,e,f} as the universe of discourse with the partition Ψ={B_1, B_2, B_3} where B_1={a,c,e},B_2={b,f} and B_3={d}. Consider the fuzzy subsets F=(0.2,0.4,0.3,0.6,0.2,0.7) and G=(0.3,0.5,0.7,0.4,0.3,0.5) of U. Then $\mu_{\underline{F}}$(B_1)=min {0.2,0.3,0.2}=0.2; $\mu_{\underline{F}}$(B_2)=min {0.4, 0.7}=0.4 and $\mu_{\underline{F}}$(B_3)=min {0.6}=0.6, which gives $\underline{F}$=(0.2,0.4,0.6). Now, $\mu_{\underline{G}}$(B_1)=min {0.3,0.7,0.3}=0.3; $\mu_{\underline{G}}$ (B_2)=min {0.5, 0.5}=0.5 and $\mu_{\underline{G}}$(B_3)=min {0.4}=0.4, which gives $\underline{G}$=(0.3,0.5,0.4). Now, F^c=(0.8,0.6,0.7,0.4,0.8,0.3) gives $\mu_{\overline{F^C}}$(B_1)=max{0.8,0.7,0.8}=0.8; $\mu_{\overline{F^C}}$(B_2)= max {0.6, 0.3}=0.6 and $\mu_{\overline{F^C}}$(B_3)= max {0.6}=0.6, which gives $\overline{F^c}$ =(0.8,0.6,0.6). Similarly, it is seen that F∪G=(0.3,0.5,0.7,0.6,0.3,0.7) and F∩G=(0.2,0.4,0.3,0.4,0.2,0.5), which gives $\underline{F \cup G}$=(0.3,0.5,0.6); $\overline{F \cup G}$=(0.7,0.7, 0.6); $\underline{F \cap G}$=(0.2,0.4,0.4) and $\overline{F \cap G}$=(0.3, 0.5,0.4).

Now, choose α=0.45. Then $\underline{F}$[α]= B_3, $\underline{G}$[α]= B_2, $\underline{F \cup G}$[α]= $B_2 \cup B_3$; $\underline{F \cap G}$[α]= Φ. Using them the properties related to lower approximations could be verified. Similarly, the other properties can be verified.

The following example shows the importance of RF-Domain in choosing the thresholds.

Example:

In the example 6.3.2, choose α=0.2. Then, $\underline{F}[\alpha]=B_2 \cup B_3$. Hence, $[\underline{F}[\alpha]]^c=B_1$. But, $\overline{F^c}[1-\alpha]=\phi$. Therefore, $[\underline{F}[\alpha]]^c \neq \overline{F^c}[1-\alpha]$, which does not satisfy 6.3.1 (f) .

In this section, it is seen that the RF-domain of the threshold α depends on the minimum and maximum membership values of the granules.

The next section compares the methods described earlier and classifies where one method is effective than another.

6.5. Comparison of de-fuzzified Rough Fuzzy set with Rough set on de-fuzzified Fuzzy Set using a threshold

In normal cases, the methods of defuzzification in two ways as mentioned above are similar. It is proved by the following theorem.

6.5.1. Theorem:

For the universe of discourse U with the partition $\Psi=\{B_1, B_2,\ldots,B_t\}$, define m_i and M_i denote the minimum and maximum of each granule B_i. Then, for any fuzzy subset A of U, by choosing a threshold α from the R-Domain, $\underline{A}[\alpha] = A_\alpha$ and $\overline{A}[\alpha] = A^\alpha$.

Proof:

Let A be any fuzzy subset of U. Now, consider an element $y \in A_\alpha$. As Z is the partition of U, $y \in X_j$ for some j. Now, $y \in A_\alpha$ implies $X_j \subseteq A[\alpha]$. Hence,

$\mu_{X_j}(x) > \alpha \ \forall x \in X_j$. Therefore, mj>α. Hence, $X_j \subseteq \underline{A}[\alpha]$, which gives, $y \in \underline{A}[\alpha]$. Therefore,

$$A_\alpha \subseteq \underline{A}[\alpha]. \qquad \textbf{(6.3)}$$

Conversely, consider an element $y \in \underline{A}[\alpha]$.As Z is the partition of U, $y \in X_j$ for some j. Therefore, $m_j > \alpha$ implies $\mu_{X_j}(x) > \alpha \ \forall x \in X_j$. Therefore, $X_j \subseteq A[\alpha]$, which gives $y \in A_\alpha$. Hence,

$$\underline{A}[\alpha] \subseteq A_\alpha. \qquad \textbf{(6.4)}$$

Hence, from 6.3 and 6.4, $\underline{A}[\alpha] = A_\alpha$. Similarly, it can be proved that $\overline{A}[\alpha] = A^\alpha$.

The above theorem is illustrated by the following example.

Example:

Consider U={a,b,c,d,e,f} as the universe of discourse with the partition $\Psi=\{B_1, B_2, B_3\}$ where B_1={a,c,e},B_2={b,f} and B_3={d}. Consider the fuzzy subset F=(0.2,0.4,0.3,0.6,0.2,0.7) of U. Then by example 6.3.2, for α=0.45, $\underline{F}[\alpha] = B_3$. Now, F[α]={d,f}. Hence, $F_\alpha = B_3$. Similarly, the equality of the upper approximations can be shown.

In general, the RF-domain gives many possibilities for the choice of the threshold. Hence, it is advisable to follow this approach rather than following the approach of R Domain. But, in the learning process, if the granulation changes from time to time, it is time-consuming to construct the RF-domain and hence, in the situations like cognition or hybridizing rough sets with the neural systems R-domain approach is more realistic and effective.

Now, the next section introduces the importance of R Domain in fuzzy predicates. Here, the approximations of fuzzy predicates are introduced and their properties are discussed.

6.6. Rough Connectives of Fuzzy Predicates

In this section, the theory of approximation discussed in the previous section is extended to fuzzy predicates.

6.6.1 Crisp representation and approximation of fuzzy predicates

For the given fuzzy predicate P, denote P{x} as the grade of membership of P(x). Then, the negation of P(x) is given by its membership function 1-P{x} and is denoted by 'neg P(x)'.

Consider the collection of fuzzy predicates $\{P_1,P_2,\ldots,P_k\}$ and the arguments $\{x_1,x_2,\ldots,x_n\}$. Let X be any partition defined on the collection of all arguments using some equivalence relation. Then P_i can be denoted as $P_i=(P_i\{x_1\}, P_i\{x_2\},\ldots, P_i\{x_n\})$. The complement of P_i is given by $P_i^c=(1-P_i\{x_1\}, 1-P_i\{x_2\},\ldots, 1-P_i\{x_n\})$. From this, it can be observed that the grades of membership of the elements of P_i^c are merely the grades of membership of the negations of $P_i(x)$.

Define the set $M=\{s/s=P_i\{x_j\} \text{ or } s=1- P_i\{x_j\}; i=1,2,\ldots,k; j=1,2,\ldots,n\}$.

Let $\alpha\in(0,1)-M$

For each α, define $P[\alpha]=\{x: P\{x\}>\alpha\}$.

The lower and upper rough approximations are defined by $P_\alpha=\underline{P[\alpha]}$ and $P^\alpha=\overline{P[\alpha]}$ respectively.

If $x \in P_\alpha$ then define $P_\alpha(x)$ is true, otherwise it is false. If $x \in P^\alpha$, then define $P^\alpha(x)$ is true otherwise it is false. As $P_\alpha \subseteq P^\alpha$, if $P_\alpha(x)$ is true then $P^\alpha(x)$ is true. Thus, for each fuzzy predicate P, the lower and upper predicates, which are crisp, can be defined with respect to α.

6.6.1.1. Result:

$P_i^c[\alpha]=\{P_i[1-\alpha]\}^c$

Proof:

$x \in P_i^c[\alpha] \Leftrightarrow 1-P_i\{x\}>\alpha \Leftrightarrow -P_i\{x\}>\alpha-1 \Leftrightarrow P_i\{x\}<1-\alpha \Leftrightarrow x \notin P_i[1-\alpha] \Leftrightarrow \{P_i[1-\alpha]\}^c$

Hence, $P_i^c[\alpha]=\{P_i[1-\alpha]\}^c$

6.6.1.2. Result:

$(P_i^c)_\alpha=(P_i^{1-\alpha})^c$

Proof:

$(P_i^c)_\alpha=\underline{[P_i^c[\alpha]]}=\underline{P_i[1-\alpha]^c}=[\overline{[P_i[1-\alpha]}]]^c=(P_i^{1-\alpha})^c$

6.6.1.3. Result:

$(P_i^c)^\alpha=(P_{i,1-\alpha})^c$

Proof:

Same as result 6.6.1.2.

From results 6.6.1.2 and 6.6.1.3, the corresponding membership values can be written as

$$(negP_i(x))_\alpha = \tau(P_i^{1-\alpha}(x))$$

$(negP_i(x))^\alpha = \tau(P_{i,1-\alpha}(x))$ where τ represents negation in usual predicate calculus.

6.6.2 Rough Connectives

In this section, the connectives are introduced similar to the connectives used in the predicate calculus.

Definition:

For the given fuzzy predicates $P_i(x)$ and $P_j(y)$, the lower and upper conjunctions $\underset{\alpha}{\wedge}$ and $\overset{\alpha}{\wedge}$ are defined as

$P_i(x) \underset{\alpha}{\wedge} P_j(y) = P_{i,\alpha}(x) \wedge P_{j,\alpha}(y)$ and **(6.5)**

$P_i(x) \overset{\alpha}{\wedge} P_j(y) = P_i^{\alpha}(x) \wedge P_j^{\alpha}(y)$ respectively. **(6.6)**

Here, $(\underset{\alpha}{\wedge}, \overset{\alpha}{\wedge})$ is called a **rough conjunction**.

Example:

Consider the universe of discourse U={a,b,c,d,e,f} with the partition Ψ={{a,c,e},{b,f},{d}}. Consider the fuzzy predicates P and Q defined on U which are given by P=(0.2,0.6,0.5,0.4,0.3,0.7) and Q=(0.4,0.6,0.3,0.6,0.5,0.8). Let α=0.45. Then P[α]={b,c,f} and Q[α]={b,d,e,f}. Hence, P_α={b,f}, P^α={a,b,c,e,f}, Q_α={b,d,f} and Q^α=Ψ.

x_1	x_2	$P(x_1)\underset{\alpha}{\wedge}P(x_2)$	$P(x_1)\overset{\alpha}{\wedge}P(x_2)$
A	A	0	1
B	A	1	1
D	A	0	0
A	B	0	1
B	B	1	1
D	B	0	0

A	D	0	1
B	D	1	1
D	d	0	0

Definition:

For the given fuzzy predicates $P_i(x)$ and $P_j(y)$, the lower and upper disjunctions $\underset{\alpha}{\vee}$ and $\overset{\alpha}{\vee}$ are defined as

$$P_i(x) \underset{\alpha}{\vee} P_j(y) = P_{i,\alpha}(x) \vee P_{j,\alpha}(y) \text{ and} \tag{6.7}$$

$$P_i(x) \overset{\alpha}{\vee} P_j(y) = P_i^{\alpha}(x) \vee P_j^{\alpha}(y) \text{ respectively} \tag{6.8}$$

Here, ($\underset{\alpha}{\vee}$, $\overset{\alpha}{\vee}$) is called **rough disjunction**.

Example:

Consider the universe of discourse U={a,b,c,d,e,f} with the partition Ψ={{a,c,e},{b,f},{d}}. Consider the fuzzy predicates P and Q defined on U which are given by P=(0.2,0.6,0.5,0.4,0.3,0.7) and Q=(0.4,0.6,0.3,0.6,0.5,0.8). Let α=0.45. Then P[α]={b,c,f} and Q[α]={b,d,e,f}. Hence, P_α={b,f}, P^α={a,b,c,e,f}, Q_α={b,d,f} and Q^α=Ψ.

x_1	x_2	$P(x_1)\underset{\alpha}{\vee}P(x_2)$	$P(x_1)\overset{\alpha}{\vee}P(x_2)$
A	A	0	1
B	A	1	1
D	A	0	1
A	B	1	1

B	B	1	1
D	B	1	1
A	D	1	1
B	D	1	1
D	D	1	1

Definition:

For the given fuzzy predicates $P_i(x)$ and $P_j(y)$, the lower and upper implications $\xrightarrow[\alpha]{}$ and $\xrightarrow{\alpha}$ are defined as

$$P_i(x) \xrightarrow[\alpha]{} P_j(y) = P_i^{1-\alpha}(x) \rightarrow P_{j,\alpha}(y) \text{ and} \quad \textbf{(6.9)}$$

$$P_i(x) \xrightarrow{\alpha} P_j(y) = P_{i,1-\alpha}(x) \rightarrow P_j^{\alpha}(y) \text{ respectively.} \quad \textbf{(6.10)}$$

Here, $(\xrightarrow[\alpha]{}, \xrightarrow{\alpha})$ is called **rough implication**.

Example:

Consider the universe of discourse U={a,b,c,d,e,f} with the partition Ψ={{a,c,e},{b,f},{d}}. Consider the fuzzy predicates P and Q defined on U which are given by P=(0.2,0.6,0.5,0.4,0.3,0.7) and Q=(0.4,0.6,0.3,0.6,0.5,0.8). Let α=0.45. Then P[α]={b,c,f} and Q[α]={b,d,e,f}. Hence, P_α={b,f}, P^α={a,b,c,e,f}, Q_α={b,d,f} and Q^α=Ψ.

x_1	x_2	$P^{1-\alpha}(x_1)$	$Q_\alpha(x_2)$	$P(x_1)\xrightarrow[\alpha]{}Q(x_2)$	$P_{1-\alpha}(x_1)$	$Q^\alpha(x_2)$	$P(x_1)\xrightarrow{\alpha}Q(x_2)$
a	A	0	0	1	0	1	1
b	A	1	0	0	1	1	1
d	A	0	0	1	0	1	1
a	B	0	1	1	0	1	1

b	B	1	1	1	1	1	1
d	B	0	1	1	0	1	1
a	D	0	1	1	0	1	1
b	D	1	1	1	1	1	1
d	D	0	1	1	0	1	1

Definition:

For the given fuzzy predicates $P_i(x)$ and $P_j(y)$, the lower and upper bi-implications $\underset{\alpha}{\longleftrightarrow}$ and $\overset{\alpha}{\longleftrightarrow}$ are defined as

$P_i(x) \underset{\alpha}{\longleftrightarrow} P_j(y) = [P_i(x) \underset{\alpha}{\longrightarrow} P_j(y)] \wedge [P_j(y) \underset{\alpha}{\longrightarrow} P_i(x)]$ and **(6.11)**

$P_i(x) \overset{\alpha}{\longleftrightarrow} P_j(y) = [P_i(x) \overset{\alpha}{\longrightarrow} P_j(y)] \wedge [P_j(y) \overset{\alpha}{\longrightarrow} P_i(x)]$ respectively. **(6.12)**

Here, ($\underset{\alpha}{\longleftrightarrow}$, $\overset{\alpha}{\longleftrightarrow}$) is called **rough bi-implication**.

Example:

Consider the universe of discourse U={a,b,c,d,e,f} with the partition Ψ={{a,c,e},{b,f},{d}}. Consider the fuzzy predicates P and Q defined on U which are given by P=(0.2,0.6,0.5,0.4,0.3,0.7) and Q=(0.4,0.6,0.3,0.6,0.5,0.8). Let α=0.45. Then P[α]={b,c,f} and Q[α]={b,d,e,f}. Hence, P_α={b,f}, P^α={a,b,c,e,f}, Q_α={b,d,f} and Q^α=Ψ.

x_1	x_2	$P(x_1)\underset{\alpha}{\leftrightarrow}Q(x_2)$	$P(x_1)\overset{\alpha}{\leftrightarrow}Q(x_2)$
a	a	1	1
b	a	0	1
d	a	1	1
a	b	0	1
b	b	1	1
d	b	0	0

a	d	0	1
b	d	1	1
d	d	0	0

Definition:

For the given fuzzy predicate $P_i(x)$, the lower and upper negations τ_α and τ^α are defined as

$$\tau_\alpha P_i(x) = (negP_i(x))_\alpha = \tau(P_i^{1-\alpha}(x)) \text{ and}$$

$$\tau^\alpha P_i(x) = (negP_i(x))^\alpha = \tau(P_{i,1-\alpha}(x)) \text{ respectively.}$$

Here, ($\tau_\alpha , \tau^\alpha$) is called **rough negation**.

Example:

Consider the universe of discourse U={a,b,c,d,e,f} with the partition Ψ={{a,c,e},{b,f},{d}}. Consider a fuzzy predicate P defined on U, which is given by P=(0.2,0.6,0.5,0.4,0.3,0.7). Let α=0.45. Then P[α]={b,c,f} gives P_α={b,f} and P^α={a,b,c,e,f}.

x_1	$\tau_\alpha P(x_1)$	$\tau^\alpha P(x_1)$
A	1	1
B	0	0
D	1	1

The above rough connectives satisfy the following properties.

6.6.2.1 *Properties*

Here, the commutative, distributive, associative properties etc., are verified.

6.6.2.1.1. *Property**:***

$$\tau\left(P_i(x) \underset{\alpha}{\vee} P_j(y)\right) = \left(negP_i(x)\right) \overset{1-\alpha}{\wedge} \left(negP_j(y)\right)$$

Proof:

$$\text{LHS} \equiv \tau\left(P_i(x) \underset{\alpha}{\vee} P_j(y)\right)$$

$$= \tau\left(P_{i,\alpha}(x) \vee P_{j,\alpha}(y)\right)$$

$$= \tau\left(P_{i,\alpha}(x)\right) \wedge \tau\left(P_{j,\alpha}(y)\right)$$

$$= \tau^{1-\alpha} P_i(x) \wedge \tau^{1-\alpha} P_j(y)$$

$$= \left(negP_i(x)\right)^{1-\alpha} \wedge \left(negP_j(y)\right)^{1-\alpha}$$

$$= \left(negP_i(x)\right) \overset{1-\alpha}{\wedge} \left(negP_j(y)\right)$$

$$\equiv \text{RHS}$$

6.6.2.1.2. *Property**:***

$$\tau\left(P_i(x) \underset{\alpha}{\wedge} P_j(y)\right) = \left(negP_i(x)\right) \overset{1-\alpha}{\vee} \left(negP_j(y)\right)$$

***Proof**:*

$$\text{LHS} \equiv \tau\left(P_i(x) \underset{\alpha}{\wedge} P_j(y)\right)$$

$$= \tau\left(P_{i,\alpha}(x) \wedge P_{j,\alpha}(y)\right)$$

$$= \tau\left(P_{i,\alpha}(x)\right) \vee \tau\left(P_{j,\alpha}(y)\right)$$

$$= \tau^{1-\alpha} P_i(x) \vee \tau^{1-\alpha} P_j(y)$$

$$= \left(negP_i(x)\right)^{1-\alpha} \vee \left(negP_j(y)\right)^{1-\alpha}$$

$$= \left(negP_i(x)\right) \overset{1-\alpha}{\vee} \left(negP_j(y)\right)$$

$\equiv$RHS

6.6.2.1.3. *Property**:***

$$\tau\left(P_i(x) \overset{\alpha}{\vee} P_j(y)\right) = \left(negP_i(x)\right) \underset{1-\alpha}{\wedge} \left(negP_j(y)\right)$$

Proof:

$$\text{LHS} \equiv \tau\left(P_i(x) \overset{\alpha}{\vee} P_j(y)\right)$$

$$= \tau\left(P_i^{\alpha}(x) \vee P_j^{\alpha}(y)\right)$$

$$= \tau\left(P_i^{\alpha}(x)\right) \wedge \tau\left(P_j^{\alpha}(y)\right)$$

$$= \tau_{1-\alpha} P_i(x) \wedge \tau_{1-\alpha} P_j(y)$$

$$= \left(negP_i(x)\right)_{1-\alpha} \wedge \left(negP_j(y)\right)_{1-\alpha}$$

$$= \left(negP_i(x)\right) \underset{1-\alpha}{\wedge} \left(negP_j(y)\right)$$

$\equiv$ RHS

6.6.2.1.4 *Property**:***

$$\tau\left(P_i(x) \overset{\alpha}{\wedge} P_j(y)\right) = \left(negP_i(x)\right) \underset{1-\alpha}{\vee} \left(negP_j(y)\right)$$

Proof:

$$\text{LHS} \equiv \tau\left(P_i(x) \overset{\alpha}{\wedge} P_j(y)\right)$$

$$= \tau\left(P_i^{\alpha}(x) \wedge P_j^{\alpha}(y)\right)$$

$$= \tau\left(P_i^{\alpha}(x)\right) \vee \tau\left(P_j^{\alpha}(y)\right)$$

$$= \tau_{1-\alpha} P_i(x) \vee \tau_{1-\alpha} P_j(y)$$

$$= \left(negP_i(x)\right)_{1-\alpha} \vee \left(negP_j(y)\right)_{1-\alpha}$$

$$= \left(negP_i(x)\right) \underset{1-\alpha}{\vee} \left(negP_j(y)\right)$$

$$\equiv \text{RHS}$$

6.6.2.1.5. *Property:*

$$P_i(x) \underset{\alpha}{\wedge} P_j(y) = P_j(y) \underset{\alpha}{\wedge} P_i(x)$$

Proof:

$$\text{LHS} \equiv P_i(x) \underset{\alpha}{\wedge} P_j(y)$$

$$= P_{i,\alpha}(x) \wedge P_{j,\alpha}(y)$$

$$= P_{j,\alpha}(y) \wedge P_{i,\alpha}(x)$$

$$= P_j(y) \underset{\alpha}{\wedge} P_i(x)$$

$$\equiv \text{RHS}$$

6.6.2.1.6. *Property**:***

$$P_i(x) \overset{\alpha}{\wedge} P_j(y) = P_j(y) \overset{\alpha}{\wedge} P_i(x)$$

Proof:

$$\text{LHS} \equiv P_i(x) \overset{\alpha}{\wedge} P_j(y)$$

$$= P_i^{\alpha}(x) \wedge P_j^{\alpha}(y)$$

$$= P_j^{\alpha}(y) \wedge P_i^{\alpha}(x)$$

$$= P_j(y) \overset{\alpha}{\wedge} P_i(x)$$

$$\equiv \text{RHS}$$

6.6.2.1.7. *Property**:***

$$P_i(x) \underset{\alpha}{\vee} P_j(y) = P_j(y) \underset{\alpha}{\vee} P_i(x)$$

Proof:

$$\text{LHS} \equiv P_i(x) \underset{\alpha}{\vee} P_j(y)$$

$$= P_{i,\alpha}(x) \vee P_{j,\alpha}(y)$$

$$= P_{j,\alpha}(y) \vee P_{i,\alpha}(x)$$

$$= P_j(y) \underset{\alpha}{\vee} P_i(x)$$

$$\equiv \text{RHS}$$

6.6.2.1.8. *Property**:***

$$P_i(x) \overset{\alpha}{\vee} P_j(y) = P_j(y) \overset{\alpha}{\vee} P_i(x)$$

Proof:

$$\text{LHS} \equiv P_i(x) \overset{\alpha}{\vee} P_j(y)$$

$$= P_i^{\alpha}(x) \vee P_j^{\alpha}(y)$$

$$= P_j^{\alpha}(y) \vee P_i^{\alpha}(x)$$

$$= P_j(y) \overset{\alpha}{\vee} P_i(x)$$

$$\equiv \text{RHS}$$

6.6.2.1.9. *Property**:***

$$P_i(x) \underset{\alpha}{\vee} [P_j(y) \underset{\alpha}{\vee} P_k(z)] = [P_i(x) \underset{\alpha}{\vee} P_j(y)] \underset{\alpha}{\vee} P_k(z)$$

Proof:

$$\text{LHS} \equiv P_i(x) \underset{\alpha}{\vee} [P_j(y) \underset{\alpha}{\vee} P_k(z)]$$

$$= P_i(x) \underset{\alpha}{\vee} [P_{j,\alpha}(y) \vee P_{k,\alpha}(z)]$$

$$= P_{i,\alpha}(x) \vee [P_{j,\alpha}(y) \vee P_{k,\alpha}(z)]$$

$$= [P_{i,\alpha}(x) \vee P_{j,\alpha}(y)] \vee P_{k,\alpha}(z)$$

$$= [P_{i,\alpha}(x) \vee P_{j,\alpha}(y)] \underset{\alpha}{\vee} P_k(z)$$

$$= [P_i(x) \underset{\alpha}{\vee} P_j(y)] \underset{\alpha}{\vee} P_k(z)$$

$$\equiv \text{RHS}$$

6.6.2.1.10. *Property**:***

$$P_i(x) \underset{\alpha}{\wedge} [P_j(y) \underset{\alpha}{\wedge} P_k(z)] = [P_i(x) \underset{\alpha}{\wedge} P_j(y)] \underset{\alpha}{\wedge} P_k(z)$$

Proof:

$$\text{LHS} \equiv P_i(x) \underset{\alpha}{\wedge} [P_j(y) \underset{\alpha}{\wedge} P_k(z)]$$

$$= P_i(x) \underset{\alpha}{\wedge} [P_{j,\alpha}(y) \wedge P_{k,\alpha}(z)]$$

$$= P_{i,\alpha}(x) \wedge [P_{j,\alpha}(y) \wedge P_{k,\alpha}(z)]$$

$$= [P_{i,\alpha}(x) \wedge P_{j,\alpha}(y)] \wedge P_{k,\alpha}(z)$$

$$= [P_{i,\alpha}(x) \wedge P_{j,\alpha}(y)] \underset{\alpha}{\wedge} P_k(z)$$

$$= [P_i(x) \underset{\alpha}{\wedge} P_j(y)] \underset{\alpha}{\wedge} P_k(z)$$

$$\equiv \text{RHS}$$

6.6.2.1.11. *Property**:***

$$P_i(x) \overset{\alpha}{\vee} [P_j(y) \overset{\alpha}{\vee} P_k(z)] = [P_i(x) \overset{\alpha}{\vee} P_j(y)] \overset{\alpha}{\vee} P_k(z)$$

Proof*:*

$$\text{LHS} \equiv P_i(x) \overset{\alpha}{\vee} [P_j(y) \overset{\alpha}{\vee} P_k(z)]$$

$$= P_i(x) \overset{\alpha}{\vee} [P_j^{\alpha}(y) \vee P_k^{\alpha}(z)]$$

$$= P_i^{\alpha}(x) \vee [P_j^{\alpha}(y) \vee P_k^{\alpha}(z)]$$

$$= [P_i^{\alpha}(x) \vee P_j^{\alpha}(y)] \vee P_k^{\alpha}(z)$$

$$= [P_i^{\alpha}(x) \vee P_j^{\alpha}(y)] \overset{\alpha}{\vee} P_k(z)$$

$$= [P_i(x) \overset{\alpha}{\vee} P_j(y)] \overset{\alpha}{\vee} P_k(z)$$

$$\equiv \text{RHS}$$

6.6.2.1.12. *Property**:***

$$P_i(x) \overset{\alpha}{\wedge} [P_j(y) \overset{\alpha}{\wedge} P_k(z)] = [P_i(x) \overset{\alpha}{\wedge} P_j(y)] \overset{\alpha}{\wedge} P_k(z)$$

Proof:

$$\text{LHS} \equiv P_i(x) \overset{\alpha}{\wedge} [P_j(y) \overset{\alpha}{\wedge} P_k(z)]$$

$$= P_i(x) \overset{\alpha}{\wedge} [P_j^{\alpha}(y) \wedge P_k^{\alpha}(z)]$$

$$= P_i^{\alpha}(x) \wedge [P_j^{\alpha}(y) \wedge P_k^{\alpha}(z)]$$

$$= [P_i^{\alpha}(x) \wedge P_j^{\alpha}(y)] \wedge P_k^{\alpha}(z)$$

$$= [P_i^{\alpha}(x) \wedge P_j^{\alpha}(y)] \overset{\alpha}{\wedge} P_k(z)$$

$$= [P_i(x) \overset{\alpha}{\wedge} P_j(y)] \overset{\alpha}{\wedge} P_k(z)$$

$$\equiv \text{RHS}$$

6.6.2.1.13. *Property**:***

$$P_i(x) \underset{\alpha}{\vee} [P_j(y) \underset{\alpha}{\wedge} P_k(z)] = [P_i(x) \underset{\alpha}{\vee} P_j(y)] \underset{\alpha}{\wedge} [P_i(x) \underset{\alpha}{\vee} P_k(z)]$$

Proof:

$$\text{LHS} \equiv P_i(x) \underset{\alpha}{\vee} [P_j(y) \underset{\alpha}{\wedge} P_k(z)]$$

$$= P_i(x) \underset{\alpha}{\vee} [P_{j,\alpha}(y) \wedge P_{k,\alpha}(z)]$$

$$= P_{i,\alpha}(x) \vee [P_{j,\alpha}(y) \wedge P_{k,\alpha}(z)]$$

$$= [P_{i,\alpha}(x) \vee P_{j,\alpha}(y)] \wedge [P_{i,\alpha}(x) \vee P_{k,\alpha}(z)]$$

$$= [P_{i,\alpha}(x) \vee P_{j,\alpha}(y)] \underset{\alpha}{\wedge} [P_{i,\alpha}(x) \vee P_{k,\alpha}(z)]$$

$$= [P_i(x) \underset{\alpha}{\vee} P_j(y)] \underset{\alpha}{\wedge} [P_i(x) \underset{\alpha}{\vee} P_k(z)]$$

$\equiv$ RHS

6.6.2.1.14. *Property**:***

$$P_i(x) \underset{\alpha}{\wedge} [P_j(y) \underset{\alpha}{\vee} P_k(z)] = [P_i(x) \underset{\alpha}{\wedge} P_j(y)] \underset{\alpha}{\vee} [P_i(x) \underset{\alpha}{\wedge} P_k(z)]$$

Proof:

$$\text{LHS} \equiv P_i(x) \underset{\alpha}{\wedge} [P_j(y) \underset{\alpha}{\vee} P_k(z)]$$

$$= P_i(x) \underset{\alpha}{\wedge} [P_{j,\alpha}(y) \vee P_{k,\alpha}(z)]$$

$$= P_{i,\alpha}(x) \wedge [P_{j,\alpha}(y) \vee P_{k,\alpha}(z)]$$

$$= [P_{i,\alpha}(x) \wedge P_{j,\alpha}(y)] \vee [P_{i,\alpha}(x) \wedge P_{k,\alpha}(z)]$$

$$= [P_{i,\alpha}(x) \wedge P_{j,\alpha}(y)] \underset{\alpha}{\vee} [P_{i,\alpha}(x) \wedge P_{k,\alpha}(z)]$$

$$= [P_i(x) \underset{\alpha}{\wedge} P_j(y)] \underset{\alpha}{\vee} [P_i(x) \underset{\alpha}{\wedge} P_k(z)]$$

$\equiv$ RHS

6.6.2.1.15. *Property**:***

$$P_i(x) \overset{\alpha}{\vee} [P_j(y) \overset{\alpha}{\wedge} P_k(z)] = [P_i(x) \overset{\alpha}{\vee} P_j(y)] \overset{\alpha}{\wedge} [P_i(x) \overset{\alpha}{\vee} P_k(z)]$$

Proof:

$$\text{LHS} \equiv P_i(x) \overset{\alpha}{\vee} [P_j(y) \overset{\alpha}{\wedge} P_k(z)]$$

$$= P_i(x) \overset{\alpha}{\vee} [P_j^{\alpha}(y) \wedge P_k^{\alpha}(z)]$$

$$= P_i^{\alpha}(x) \vee [P_j^{\alpha}(y) \wedge P_k^{\alpha}(z)]$$

$$= [P_i^{\alpha}(x) \vee P_j^{\alpha}(y)] \wedge [P_i^{\alpha}(x) \vee P_k^{\alpha}(z)]$$

$$= [P_i^{\alpha}(x) \vee P_j^{\alpha}(y)] \overset{\alpha}{\wedge} [P_i^{\alpha}(x) \vee P_k^{\alpha}(z)]$$

$$= [P_i(x) \overset{\alpha}{\vee} P_j(y)] \overset{\alpha}{\wedge} [P_i(x) \overset{\alpha}{\vee} P_k(z)]$$

$\equiv$ RHS

6.6.2.1.16. *Property**:***

$$P_i(x) \overset{\alpha}{\wedge} [P_j(y) \overset{\alpha}{\vee} P_k(z)] = [P_i(x) \overset{\alpha}{\wedge} P_j(y)] \overset{\alpha}{\vee} [P_i(x) \overset{\alpha}{\wedge} P_k(z)]$$

Proof:

LHS $\equiv P_i(x) \overset{\alpha}{\wedge} [P_j(y) \overset{\alpha}{\vee} P_k(z)]$

$$= P_i(x) \overset{\alpha}{\wedge} [P_j^{\alpha}(y) \vee P_k^{\alpha}(z)]$$

$$= P_i^{\alpha}(x) \wedge [P_j^{\alpha}(y) \vee P_k^{\alpha}(z)]$$

$$= [P_i^{\alpha}(x) \wedge P_j^{\alpha}(y)] \vee [P_i^{\alpha}(x) \wedge P_k^{\alpha}(z)]$$

$$= [P_i^{\alpha}(x) \wedge P_j^{\alpha}(y)] \overset{\alpha}{\vee} [P_i^{\alpha}(x) \wedge P_k^{\alpha}(z)]$$

$$= [P_i(x) \overset{\alpha}{\wedge} P_j(y)] \overset{\alpha}{\vee} [P_i(x) \overset{\alpha}{\wedge} P_k(z)]$$

$\equiv$ RHS

6.6.2.1.17. *Property:*

$$P_i(x) \xrightarrow[\alpha]{} P_j(y) = (negP_i(x)) \underset{\alpha}{\vee} P_j(y)$$

Proof:

LHS $\equiv P_i(x) \xrightarrow[\alpha]{} P_j(y)$

$$= P_i^{1-\alpha}(x) \to P_{j,\alpha}(y)$$

$$= \tau P_i^{1-\alpha}(x) \vee P_{j,\alpha}(y)$$

$$= \tau_{\alpha} P_i(x) \vee P_{j,\alpha}(y)$$

$$= (negP_i(x))_{\alpha} \vee P_{j,\alpha}(y)$$

$$= (negP_i(x)) \underset{\alpha}{\vee} P_j(y)$$

$\equiv$ RHS

6.6.2.1.18. *Property:*

$$P_i(x) \xrightarrow{\alpha} P_j(y) = (negP_i(x)) \overset{\alpha}{\vee} P_j(y)$$

Proof:

$$\text{LHS} \equiv P_i(x) \xrightarrow{\alpha} P_j(y)$$

$$= P_{i,1-\alpha}(x) \rightarrow P_j^{\alpha}(y)$$

$$= \tau P_{i,1-\alpha}(x) \vee P_j^{\alpha}(y)$$

$$= \tau^{\alpha} P_i(x) \vee P_j^{\alpha}(y)$$

$$= (negP_i(x))^{\alpha} \vee P_j^{\alpha}(y)$$

$$= (negP_i(x)) \overset{\alpha}{\vee} P_j(y)$$

$$\equiv \text{RHS}$$

6.6.2.1.19. *Property**:***

$$\tau[P_i(x) \xrightarrow[\alpha]{} P_j(y)] = P_i(x) \overset{1-\alpha}{\wedge} (neg(P_j(y))$$

Proof:

$$\text{LHS} \equiv \tau[P_i(x) \xrightarrow[\alpha]{} P_j(y)]$$

$$= \tau[P_i^{1-\alpha}(x) \rightarrow P_{j,\alpha}(y)]$$

$$= \tau[\tau P_i^{1-\alpha}(x) \vee P_{j,\alpha}(y)]$$

$$= P_i^{1-\alpha}(x) \wedge \tau P_{j,\alpha}(y)$$

$$= P_i^{1-\alpha}(x) \wedge \tau^{1-\alpha} P_j(y)$$

$$= P_i^{1-\alpha}(x) \wedge (negP_j(y))^{1-\alpha}$$

$$= P_i(x) \overset{1-\alpha}{\wedge} (neg(P_j(y))$$

$$\equiv \text{RHS}$$

6.6.2.1.20. *Property**:***

$$\tau[P_i(x) \xrightarrow{\alpha} P_j(y)] = P_i(x) \underset{1-\alpha}{\wedge} (neg(P_j(y))$$

Proof:

$$\text{LHS} \equiv \tau[P_i(x) \xrightarrow{\alpha} P_j(y)]$$

$$= \tau[P_{i,1-\alpha}(x) \rightarrow P_j^{\alpha}(y)]$$

$$= \tau[\tau P_{i,1-\alpha}(x) \vee P_j^{\alpha}(y)]$$

$$= P_{i,1-\alpha}(x) \wedge \tau P_j^{\alpha}(y)$$

$$= P_{i,1-\alpha}(x) \wedge \tau_{1-\alpha} P_j(y)$$

$$= P_{i,1-\alpha}(x) \wedge (negP_j(y))_{1-\alpha}$$

$$= P_i(x) \underset{1-\alpha}{\wedge} (neg(P_j(y))$$

$$\equiv \text{RHS}$$

6.6.2.1.21. *Property**:***

$$P_i(x) \underset{\alpha}{\wedge} (negP_i(x)) = .\text{f}.$$

Proof:

$$P_i(x) \underset{\alpha}{\wedge} (negP_i(x)) = P_{i,\alpha} \wedge (negP_i(x))_{\alpha} = \; P_{i,\alpha}(x) \wedge \tau_{\varepsilon} P_i(x) = P_{i,\alpha}(x) \wedge \tau P_i^{1-\alpha}(x) .$$

If $P_{i,\alpha}(x)$ is true then

$x \in (P_i)_{\alpha} \Rightarrow x \notin [(P_i)_{\alpha}]^c \Rightarrow x \notin [(P_i)^c]^{1-\alpha} \Rightarrow \tau P_i^{1-\alpha}(x)$ is false. Similarly, when $P_{i,\alpha}(x)$ is false, then $\tau P_i^{1-\alpha}(x)$ is true. Hence, the result follows.

Similarly, the following results can be proved:

$$P_i(x) \overset{\alpha}{\wedge} (negP_i(x)) = .f.$$

$$P_i(x) \underset{\alpha}{\vee} (negP_i(x)) = .t.$$

$$P_i(x) \overset{\alpha}{\vee} (negP_i(x)) = .t.$$

Here, the rough tools are derived for connectives of fuzzy predicates. These rough connectives help in making decision rules on the fuzzy system. Additionally, the other rules of inferences can also be derived.

Now, we concentrate on implementing the above concepts into three regions, namely Positive, Negative and Boundary Regions introduced by Pawlak.

6.7 Analysis of the regions of RS-Model using thresholds under Fuzziness

As discussed earlier, the boundary BND(A) is defined as BND(A) = $\overline{A}$ - $\underline{A} \neq \phi$. Hence if BND(A) = ϕ, then A is said to be exact. Any object in $\underline{A}$ gives the certainty of the object in X with respect to R. Any object in $\overline{A}$ gives the possibility of the object in X with respect to R. Hence $\underline{A}$ is termed as the positive region of A which is meant by POS(A) and U- $\overline{A}$ is termed as the negative region of A, meant by NEG(A).

By using the notion of thresholds in fuzzy sets and the above said basics the regions of rough sets can be modified as follows:

Let U be the universe of discourse and X be the partition of U. Let A be any fuzzy subset of U and α be the threshold taken from R- domain. Then define the following three regions of rough sets with respect to α as follows:

Positive-region: $POS_\alpha(A)=A_\alpha$

Negative region: $NEG_\alpha(A)=U- A^\alpha$

$=(A^\alpha)^c$

$=(A^c)_{1-\alpha}$

Boundary Region: $BND_\alpha(A)= A^\alpha - A_\alpha$

$=A^\alpha \cap (A_\alpha)^c$

$=A^\alpha \cap (A^c)^{1-\alpha}$

6.7.1 Analysis of the regions of RS-Model using the threshold α

Let A and B be two fuzzy subsets of U and α be the threshold. Then the following properties are derived.

6.7.1.1 *Property**:***

$POS_{\alpha}(A) \cup POS_{\alpha}(B) \subseteq POS_{\alpha}(A \cup B)$

Proof:

$POS_{\alpha}(A) \cup POS\alpha(B) = A_{\alpha} \cup B_{\alpha} \subseteq (A \cup B)_{\alpha} = POS_{\alpha}(A \cup B)$

6.7.1.2 *Property**:***

$POS_{\alpha}(A) \cap POS_{\alpha}B = POS_{\alpha}(A \cap B)$

***Proof*:**

$POS_{\alpha}(A) \cap POS\alpha(B) = A_{\alpha} \cap B_{\alpha} = (A \cap B)_{\alpha} = POS_{\alpha}(A \cap B)$

6.7.1.3 *Property**:***

$NEG_{\alpha}(A) \cap NEG_{\alpha}(B) = NEG_{\alpha}(A \cup B)$

***Proof*:**

$$
\begin{aligned}
NEG_{\alpha}(A) \cap NEG_{\alpha}(B) &= (A^{c})_{1-\alpha} \cap (B^{c})_{1-\alpha} \\
&= (A^{c} \cap B^{c})_{1-\alpha} \\
&= [(A \cup B)^{c}]_{1-\alpha} \\
&= NEG_{\alpha}(A \cup B)
\end{aligned}
$$

6.7.1.4 *Property:*

$NEG_{\alpha}(A) \cup NEG_{\alpha}(B) \subseteq NEG_{\alpha}(A \cap B)$

Proof:

$$NEG_{\alpha}(A)\cup NEG_{\alpha}(B) =(A^{c})_{1-\alpha}\cup(B^{c})_{1-\alpha}$$
$$\subseteq(A^{c}\cup B^{c})_{1-\alpha}$$
$$=[(A\cap B)^{c}]_{1-\alpha}$$
$$=NEG_{\alpha}(A\cap B)$$

6.7.1.5 *Property**:***

$$BND_{\alpha}(A)\cup BND_{\alpha}(B) \subseteq [NEG_{\alpha}(A\cup B)]^{c}\cap[POS_{\alpha}(A\cap B)]^{c}$$

Proof:

$$BND_{\alpha}(A)\cup BND_{\alpha}(B) =[A^{\alpha} \cap(A^{c})^{1-\alpha}]\cup[B^{\alpha} \cap (B^{c})^{1-\alpha}]$$
$$=\{[A^{\alpha} \cap(A^{c})^{1-\alpha}]\cup B^{\alpha}\}\cap\{[A^{\alpha} \cap(A^{C})^{1-\alpha}] \cup(B^{C})^{1-\alpha}\}$$
$$=[A^{\alpha} \cup B^{\alpha}]\cap[(A^{C})^{1-\alpha} \cup B^{\alpha}]\cap[A^{\alpha} \cup(B^{C})^{1-\alpha}]\cap[(A^{C})^{1\alpha} \cup(B^{C})^{1-\alpha}]$$
$$\subseteq[A^{\alpha} \cup B^{\alpha}] \cap [(A^{c})^{1-\alpha} \cup (B^{c})^{1-\alpha}]$$
$$=(A\cup B)^{\alpha} \cap (A^{c} \cup B^{c})^{1-\alpha}$$
$$=(A\cup B)^{\alpha} \cap [(A\cap B)^{c}]^{1-\alpha}$$
$$=(A\cup B)^{\alpha} \cap [(A\cap B)_{\alpha}]^{c}$$
$$=[NEG_{\alpha}(A\cup B)]^{c}\cap[POS_{\alpha}(A\cap B)]^{c}$$

6.7.1.6 *Property**:***

$$BND_{\alpha}(A)\cap BND_{\alpha}(B)\supseteq[NEG_{\alpha}(A\cap B)]^{c}\cap[POS_{\alpha}(A\cup B)]^{c}$$

Proof:

$$BND_{\alpha}(A)\cap BND_{\alpha}(B) =[A^{\alpha} \cap (A^{c})^{1-\alpha}] \cap [B^{\alpha} \cap (B^{c})^{1-\alpha}]$$
$$=(A^{\alpha} \cap B^{\alpha}) \cap [(A^{c})^{1-\alpha}\cap(B^{c})^{1-\alpha}]$$
$$\supseteq(A\cap B)^{\alpha}\cap(A^{c}\cap B^{c})^{1-\alpha}$$

$=(A\cap B)^{\alpha}\cap[(A\cup B)^{c}]^{1-\alpha}$

$=(A\cap B)^{\alpha}\cap[(A\cup B)_{\alpha}]^{c}$

$=[NEG_{\alpha}(A\cap B)]^{c}\cap[POS_{\alpha}(A\cup B)]^{c}$

6.7.1.7 *Property**:***

$[BND_{\alpha}(A)]^{c}=NEG_{\alpha}(A)\cup POS_{\alpha}(A)$

***Proof*:**

$[BND_{\alpha}(A)]^{c}=[A^{\alpha}\cap(A^{c})^{1-\alpha}]^{c}$

$=(A^{\alpha})^{c}\cup[(A^{c})^{1-\alpha}]^{c}$

$=(A^{\alpha})^{c}\cup[(A^{c})^{c}]_{\alpha}$

$=(A^{\alpha})^{c}\cup(A)_{\alpha}$

$=NEG_{\alpha}(A)\cup POS_{\alpha}(A)$

The above properties can be illustrated as follows:

*Example**:***

Let U={a,b,c,d,e} be the universe of discourse and X={{a,b,d},{c},{e}} be the partition of U. Let A=(0.6,0.8,0.4,0.2,1), B=(0.5,0.3,0.6,0.5,0) be two fuzzy subsets of U. Then D=(0,1)- {0.2,0.3,0.4,0.5,0.6,0.7,0.8}, $\alpha\in D$ say $\alpha=0.45$.

Then A[α]={a,b,e}; B[α]={a,c,d}. Hence A_{α}={e}; A^{α}={a,b,d,e}; B_{α}={c}; B^{α}={a,b,c,d}

$A\cup B$=(0.6,0.8,0.6,0.5,1) ; $A\cap B$=(0.5,0.3,0.4,0.2,0); A^{c}=(0.4,0.2,0.6,0.8,0)

$A\cup B$[α]={a,b,c,d,e}; $A\cap B$[α]={a}; A^{c}[1-α]={c,d}.

Therefore, $(A\cup B)^{\alpha}=U=(A\cup B)_{\alpha}$; $(A\cap B)_{\alpha}=\Phi$; $(A\cap B)^{\alpha}$={a,b,d}; $(A^{c})_{1-\alpha}$={c}; $(A^{c})^{1-\alpha}$={a,b,c,d}. Hence $POS_{\alpha}(A)$={e}; $NEG_{\alpha}(A)$=U-{a,b,d,e}={c};

$BND_\alpha(A)=\{a,b,d,e\}-\{e\}=\{a,b,d\}$;$POS_\alpha(B)=\{c\}$; $NEG_\alpha(B)=U-\{a,b,c,d\}=\{e\}$; $BND_\alpha(B)=\{a,b,c,d\}-\{c\}=\{a,b.d\}$; $POS_\alpha(A\cup B)=\{a,b,c,d,e]$; $NEG_\alpha(A\cup B) = \Phi$; $POS_\alpha(A\cap B) = \Phi$; $NEG_\alpha(A\cap B) = U-\{a,b,d\}=\{c,e\}$.Hence,

1)$POS_\alpha (A)\cup POS\alpha(B)=\{e\}\cup\{c\}=\{c,e\}\subseteq\{a,b,c,d,e\}=POS_\alpha(A\cup B)$

2) $POS_\alpha (A)\cap POS\alpha(B)=\{e\}\cap\{c\}=\Phi=POS_\alpha(A\cap B)$

3) $NEG_\alpha(A)\cap NEG_\alpha(B)= \Phi=NEG_\alpha(A\cup B)$

4) $NEG_\alpha(A)\cup NEG_\alpha(B)=\{c\}\cup\{e\}=\{c,e\}=NEG_\alpha(A\cap B)$

5) $BND_\alpha(A)\cup BND_\alpha(B)=\{a,b,d\}\cup\{a,b,d\}=\{a,b,d\}$

$[NEG_\alpha(A\cup B)]^c=U$; $[POS_\alpha(A\cap B)]^c=U$. Hence $[NEG_\alpha(A\cup B)]^c\cap[POS_\alpha(A\cap B)]^c=U$

Therefore, $BND_\alpha(A)\cup BND_\alpha(B)\subseteq[NEG_\alpha(A\cup B)]^c\cap[POS_\alpha(A\cap B)]^c$

6) $BND_\alpha(A)\cap BND_\alpha(B)=\{a,b,d\}$

$[NEG_\alpha(A\cap B)]^c=\{a,b,d\}$; $[POS_\alpha(A\cup B)]^c=\Phi$. Hence $[NEG_\alpha(A\cap B)]^c\cap[POS_\alpha(A\cup B)]^c= \Phi$

Therefore, $BND_\alpha(A)\cap BND_\alpha(B)\supseteq[NEG_\alpha(A\cap B)]^c\cap[POS_\alpha(A\cup B)]^c$

7) $[BND_\alpha(A)]^c =U-\{a,b,d\}=\{c,e\}$

$NEG_\alpha(A)\cup POS_\alpha(A) =\{c\}\cup\{e\}=\{c,e\}$. Therefore, $[BND_\alpha(A)]^c =NEG_\alpha(A)\cup POS_\alpha(A)$

6.7.2 Analysis of the regions of RS-Model using the thresholds α_1 and α_2

If α_1 and α_2 are two thresholds, then, $A[\alpha_1]\cup A[\alpha_2]=A[\alpha]$, where $\alpha=\min(\alpha_1,\alpha_2)$ and $A[\alpha_1]\cap A[\alpha_2]=A[\alpha]$, where $\alpha=\max(\alpha_1,\alpha_2)$. Then, trivially, $A_{\alpha_1} \cup A_{\alpha_2} = A_\alpha$ where $\alpha=\min(\alpha_1,\alpha_2)$ and $A_{\alpha_1} \cap A_{\alpha_2} = A_\alpha$ where $\alpha=\max(\alpha_1,\alpha_2)$. By using these results the following properties are derived:

***6.7.2.1** Property:*

$POS_{\alpha_1}(A) \cup POS_{\alpha_2}(A) = POS_{\alpha}(A)$ Where α=min(α_1,α_2)

***Proof*:**

$POS_{\alpha_1}(A) \cup POS_{\alpha_2}(A) \quad = A_{\alpha_1} \cup A_{\alpha_2}$

$=A_{\alpha}$ where α=min(α_1,α_2)

$=POS_{\alpha}$ (A) Where α=min(α_1,α_2)

***6.7.2.2** Property:*

$POS_{\alpha_1}(A) \cap POS_{\alpha_2}(A) = POS_{\alpha}(A)$ where α=max(α_1,α_2)

***Proof*:**

$POS_{\alpha_1}(A) \cap POS_{\alpha_2}(A) \quad = A_{\alpha_1} \cap A_{\alpha_2}$

$=A_{\alpha}$ where α=max(α_1,α_2)

$=POS_{\alpha}(A)$ where α=max(α_1,α_2)

***6.7.2.3** Property:*

$NEG_{\alpha_1}(A) \cup NEG_{\alpha_2}(A) = NEG_{\alpha}(A)$ where α=max(α_1,α_2)

***Proof*:**

$NEG_{\alpha_1}(A) \cup NEG_{\alpha_2}(A) \quad = (A^{\alpha_1})^c \cup (A^{\alpha_2})^c$

$= (A^{\alpha_1} \cap A^{\alpha_2})^c$

$= (A^{\alpha})^c$ where α=max(α_1,α_2)

$=NEG_{\alpha}(A)$ where α=max(α_1,α_2)

***6.7.2.4** Property:*

$NEG_{\alpha_1}(A) \cap NEG_{\alpha_2}(A) = NEG_{\alpha}(A)$ where α = min (α_1,α_2)

***Proof*:**

$$NEG_{\alpha_1}(A) \cap NEG_{\alpha_2}(A) = (A^{\alpha_1})^c \cap (A^{\alpha_2})^c$$

$$= (A^{\alpha_1} \cup A^{\alpha_2})^c$$

$= (A^{\alpha})^c$ where $\alpha=\min(\alpha_1,\alpha_2)$

$=NEG_{\alpha}(A)$ where $\alpha=\min(\alpha_1,\alpha_2)$

6.7.2.5 *Property****:***

$BND_{\alpha_1}(A) \cup BND_{\alpha_2}(A) \subseteq [NEG_{\alpha}(A)]^c \cap [POS_{\beta}(A)]^c$ where $\alpha=\min(\alpha_1,\alpha_2)$; $\beta=\max(\alpha_1,\alpha_2)$

***Proof*:**

$$BND_{\alpha_1}(A) \cup BND_{\alpha_2}(A) = [A^{\alpha_1} \cap (A_{\alpha_1})^c] \cup [A^{\alpha_2} \cap (A_{\alpha_2})^c]$$

$$= \{[A^{\alpha_1} \cap (A_{\alpha_1})^c] \cup A^{\alpha_2}\} \cap \{[A^{\alpha_1} \cap (A_{\alpha_1})^c] \cup (A_{\alpha_2})^c\}$$

$$= (A^{\alpha_1} \cup A^{\alpha_2}) \cap [(A_{\alpha_1})^c \cup A^{\alpha_2}] \cap [A^{\alpha_1} \cup (A_{\alpha_2})^c] \cap [(A_{\alpha_1})^c \cup (A_{\alpha_2})^c]$$

$$\subseteq [(A^{\alpha_1} \cup A^{\alpha_2})] \cap [(A_{\alpha_1})^c \cup (A_{\alpha_2})^c]$$

$$= (A^{\alpha_1} \cup A^{\alpha_2}) \cap (A_{\alpha_1} \cap A_{\alpha_2})^c$$

$=(A)^{\alpha} \cap (A_{\beta})^C$ where $\alpha = \min(\alpha_1,\alpha_2)$; $\beta = \max(\alpha_1,\alpha_2)$

$=[NEG_{\alpha}(A)]^c \cap [POS_{\beta}(A)]^c$ where $\alpha=\min(\alpha_1,\alpha_2)$; $\beta=\max(\alpha_1,\alpha_2)$

6.7.2.6 *Property****:***

$BND_{\alpha_1}(A) \cap BND_{\alpha_2}(A) = [NEG_{\alpha}(A)]^c \cap [POS_{\beta}(A)]^c$ where $\alpha=\max(\alpha_1,\alpha_2)$; $\beta=\min(\alpha_1,\alpha_2)$

***Proof*:**

$$BND_{\alpha_1}(A) \cap BND_{\alpha_2}(A) = [A^{\alpha_1} \cap (A_{\alpha_1})^c] \cap [A^{\alpha_2} \cap (A_{\alpha_2})^c]$$

$=(A^{\alpha_1} \cap A^{\alpha_2}) \cap [(A_{\alpha_1})^c \cap (A_{\alpha_2})^c]$

$=(A^{\alpha_1} \cap A^{\alpha_2}) \cap (A_{\alpha_1} \cup A_{\alpha_2})^c$

$=A^{\alpha} \cap (A_{\beta})^c$ where $\alpha=\max(\alpha_1,\alpha_2)$; $\beta=\min(\alpha_1,\alpha_2)$

$=[NEG_{\alpha}(A)]^c \cap [POS_{\beta}(A)]^c$ where $\alpha=\max(\alpha_1,\alpha_2)$; $\beta=\min(\alpha_1,\alpha_2)$

6.7.2.7 *Property**:***

$$POS_{\alpha_1}(A) \cup POS_{\alpha_2}(B) \subseteq POS_{\alpha}(A \cup B) \text{ where } \alpha = \min(\alpha_1,\alpha_2)$$

***Proof*:**

$POS_{\alpha_1}(A) \cup POS_{\alpha_2}(B) = A_{\alpha_1} \cup B_{\alpha_2}$

$\subseteq A_{\alpha} \cup B_{\alpha}$ where $\alpha = \min(\alpha_1,\alpha_2)$

$\subseteq (A \cup B)_{\alpha}$ where $\alpha = \min(\alpha_1,\alpha_2)$

$=POS_{\alpha}(A \cup B)$ where $\alpha = \min(\alpha_1,\alpha_2)$

6.7.2.8 *Property**:***

$$POS_{\alpha_1}(A) \cap POS_{\alpha_2}(B) \supseteq POS_{\alpha}(A \cap B) \text{ where } \alpha=\max(\alpha_1,\alpha_2)$$

***Proof*:**

$POS_{\alpha_1}(A) \cap POS_{\alpha_2}(B) = A_{\alpha_1} \cap B_{\alpha_2}$

$\supseteq A_{\alpha} \cap B_{\alpha}$ where $\alpha=\max(\alpha_1,\alpha_2)$

$=(A \cap B)_{\alpha}$ where $\alpha=\max(\alpha_1,\alpha_2)$

$=POS_{\alpha}(A \cap B)$ where $\alpha=\max(\alpha_1,\alpha_2)$

6.7.2.9 *Property**:***

$$NEG_{\alpha_1}(A) \cup NEG_{\alpha_2}(B) \subseteq NEG_{\alpha}(A \cap B) \text{ where } \alpha=\max(\alpha_1,\alpha_2)$$

***Proof*:**

$NEG_{\alpha_1} A \cup NEG_{\alpha_2} B = (A^{\alpha_1})^c \cup (B^{\alpha_2})^c$

$=(A^c)_{1-\alpha_1} \cup (B^c)_{1-\alpha_2}$

$\subseteq(A^c)_{1-\alpha}\cup(B^c)_{1-\alpha}$ where $\alpha=\max(\alpha_1,\alpha_2)$

$\subseteq(A^c \cup B^c)_{1-\alpha}$ where $\alpha=\max(\alpha_1,\alpha_2)$

$=[(A\cap B)^c]_{1-\alpha}$ where $\alpha=\max(\alpha_1,\alpha_2)$

$= [(A\cap B)^\alpha]^c$ where $\alpha=\max(\alpha_1,\alpha_2)$

$= NEG_\alpha(A\cap B)$ where $\alpha=\max(\alpha_1,\alpha_2)$

***6.7.2.10** Property:*

$$NEG_{\alpha_1}(A) \cap NEG_{\alpha_2}(B) \supseteq NEG_\alpha(A \cup B) \text{ where } \alpha=\min(\alpha_1,\alpha_2)$$

***Proof*:**

$NEG_{\alpha_1}(A) \cap NEG_{\alpha_2}(B) = (A^{\alpha_1})^c \cap (B^{\alpha_2})^c$

$=(A^c)_{1-\alpha_1} \cap (B^c)_{1-\alpha_2}$

$\supseteq(A^c)_{1-\alpha} \cap(B^c)_{1-\alpha}$ where $\alpha=\min(\alpha_1,\alpha_2)$

$= (A^c\cap B^c)_{1-\alpha}$ where $\alpha=\min(\alpha_1,\alpha_2)$

$=[(A\cup B)^c]_{1-\alpha}$ where $\alpha=\min(\alpha_1,\alpha_2)$

$=[(A\cup B)^\alpha]^c$ where $\alpha=\min(\alpha_1,\alpha_2)$

$=NEG_\alpha(A\cup B)$ where $\alpha=\min(\alpha_1,\alpha_2)$

***6.7.2.11** Property:*

$BND_{\alpha_1}(A) \cup BND_{\alpha_2}(B) \subseteq [NEG_\alpha(A \cup B)]^c \cap [POS_\beta(A \cap B)]^c$ where $\alpha = \min(\alpha_1,\alpha_2)$; $\beta = \max(\alpha_1,\alpha_2)$

***Proof*:**

$BND_{\alpha_1}(A) \cup BND_{\alpha_2}(B) = [A^{\alpha_1} \cap (A_{\alpha_1})^c] \cup [B^{\alpha_2} \cap (B_{\alpha_2})^c]$

$=[A^{\alpha_1} \cap (A^c)^{1-\alpha_1}] \cup [B^{\alpha_2} \cap (B^c)^{1-\alpha_2}]$

$=\{[A^{\alpha_1} \cap (A^c)^{1-\alpha_1}] \cup B^{\alpha_2}\} \cap \{[A^{\alpha_1} \cap (A^c)^{1-\alpha_{1}}] \cup (B^c)^{1-\alpha_2}\}$

$=(A^{\alpha_1}\cup B^{\alpha_2})\cap[(A^c)^{1-\alpha_1}\cup B^{\alpha_2}]\cap[A^{\alpha_1}\cup(B^c)^{1-\alpha_2}]\cap[(A^c)^{1-\alpha_1}\cup(B^c)^{1-\alpha_2}]$

$\subseteq(A^{\alpha_1}\cup B^{\alpha_2})\cap[(A^c)^{1-\alpha_1}\cup(B^c)^{1-\alpha_2}]$

$\subseteq(A^{\alpha}\cup B^{\alpha})\cap[(A^c)^{1-\beta}\cup(B^c)^{1-\beta}]$ where $\alpha=\min(\alpha_1,\alpha_2)$; $\beta=\max(\alpha_1,\alpha_2)$

$=(A\cup B)^{\alpha}\cap(A^c\cup B^c)^{1-\beta}$ where $\alpha=\min(\alpha_1,\alpha_2)$; $\beta=\max(\alpha_1,\alpha_2)$

$=(A\cup B)^{\alpha}\cap[(A\cap B)^c]^{1-\beta}$ where $\alpha=\min(\alpha_1,\alpha_2)$; $\beta=\max(\alpha_1,\alpha_2)$

$=(A\cup B)^{\alpha}\cap[(A\cap B)_{\beta}]^c$ where $\alpha=\min(\alpha_1,\alpha_2)$; $\beta=\max(\alpha_1,\alpha_2)$

$=[NEG_{\alpha}(A\cup B)]^c\cap[POS_{\beta}(A\cap B)]^c$ where $\alpha=\min(\alpha_1,\alpha_2)$; $\beta=\max(\alpha_1,\alpha_2)$

6.7.2.12 *Property**:***

$BND_{\alpha_1}(A)\cap BND_{\alpha_2}(B)\supseteq[NEG_{\alpha}(A\cap B)]^c\cap[POS_{\beta}(A\cup B)]^c$ where $\alpha=\max(\alpha_1,\alpha_2)$; $\beta=\min(\alpha_1,\alpha_2)$

***Proof*:**

$BND_{\alpha_1}(A)\cap BND_{\alpha_2}(B)=[A^{\alpha_1}\cap(A_{\alpha_1})^c]\cap[B^{\alpha_2}\cap(B_{\alpha_2})^c]$

$=[A^{\alpha_1}\cap(A^c)^{1-\alpha_1}]\cap[B^{\alpha_2}\cap(B^c)^{1-\alpha_2}]$

$=(A^{\alpha_1}\cap B^{\alpha_2})\cap[(A^c)^{1-\alpha_1}\cap(B^c)^{1-\alpha_2}]$

$\supseteq(A^{\alpha}\cap B^{\alpha})\cap[(A^c)^{1-\beta}\cap(B^c)^{1-\beta}]$ where $\alpha=\max(\alpha_1,\alpha_2)$; $\beta=\min(\alpha_1,\alpha_2)$

$\supseteq(A\cap B)^{\alpha}\cap(A^c\cap B^c)^{1-\beta}$ where $\alpha=\max(\alpha_1,\alpha_2)$; $\beta=\min(\alpha_1,\alpha_2)$

$=(A\cap B)^{\alpha}\cap[(A\cup B)^c]^{1-\beta}$ where $\alpha=\max(\alpha_1,\alpha_2)$; $\beta=\min(\alpha_1,\alpha_2)$

$=(A\cap B)^{\alpha}\cap[(A\cup B)_{\beta}]^c$ where $\alpha=\max(\alpha_1,\alpha_2)$; $\beta=\min(\alpha_1,\alpha_2)$

$=[NEG_{\alpha}(A\cap B)]^{c}\cap[POS_{\beta}(A\cup B)]^{c}$ where α=max(α1,α2); β=min(α1,α2).

The following example illustrates the above properties.

Example:

Consider U={a,b,c,d,e} be the universe of discourse and X={{a,c},{b},{d,e}} be the partition of U. Suppose that A=(0.2,0.5,0.4,0.8,0.6) and B=(0.7,0.2,0.8,0,0.5) are fuzzy subsets of U. Let α_1=0.45 and α_2=0.65 be two thresholds. Hence in this case $\alpha_1 < \alpha_2$ which implies $\min(\alpha_1,\alpha_2)=\alpha_1$ and $\max(\alpha_1,\alpha_2)=\alpha_2$

Then $A[\alpha_1]$ ={b,d,e}; $A[\alpha_2]$={d}; $B[\alpha_2]$={a,c}.Hence $A_{\alpha_1}=\{b,d,e\}=A^{\alpha_1}$; $A_{\alpha_2}=\Phi; A^{\alpha_2}=\{d,e\}$

$B_{\alpha_2}=\{a,c\}=B^{\alpha_2}$.Now,$(A\cup B)[\alpha_1]$={a,b,c,d,e} and $(A\cap B)[\alpha_2]=\Phi$.

Hence $(A\cup B)_{\alpha_1}=(A\cup B)^{\alpha_1}=U$ and $(A\cap B)_{\alpha_1}=(A\cap B)^{\alpha_1}=\Phi$.

Therefore, $POS_{\alpha_1}(A)=\{b,d,e\}; NEG_{\alpha_1}(A)=\{a,c\}; BND_{\alpha_1}(A)=\Phi$;

$POS_{\alpha_2}(A)=\Phi;\ NEG_{\alpha_2}(A)=\{a,b,c\}; BND_{\alpha_2}(A)=\{d,e\}$;

$POS_{\alpha_2}(B)=\{a,c\}; NEG_{\alpha_2}(B)=\{b,d,c\}; BND_{\alpha_2}(B)=\Phi$;

$POS_{\alpha_1}(A\cup B)=U$; $NEG_{\alpha_1}(A\cup B)=\Phi$; $BND_{\alpha_1}(A\cup B)=\Phi$;

$POS_{\alpha_2}(A\cap B)=\Phi$; $NEG_{\alpha_2}(A\cap B)=U$; $BND_{\alpha_2}(A\cap B)=\Phi$.

Then the above properties can be verified as follows:

1) $POS_{\alpha_1}(A)\cup POS_{\alpha_2}(A)$={b,d,e}=$POS_{\alpha_1}(A)$
2) $POS_{\alpha_1}(A)\cap POS_{\alpha_2}(A)=\Phi=POS_{\alpha_2}(A)$
3) $NEG_{\alpha_1}(A)\cup NEG_{\alpha_2}(A)$={a,b,c} $=NEG_{\alpha_2}(A)$

4) $NEG_{\alpha_1}(A) \cap NEG_{\alpha_2}(A)$ ={a,c}= $NEG_{\alpha_1}(A)$

5) $BND_{\alpha_1}(A) \cup BND_{\alpha_2}(A)$ ={d,e}

$[NEG_{\alpha_1}(A)]^c$ ={b,d,e}, $[POS_{\alpha_2}(A)]^c$ ={a,b,c,d,e}

$[NEG_{\alpha_1}(A)]^c \cap [POS_{\alpha_2}(A)]^c$ ={b,d,e}

Therefore, $BND_{\alpha_1}(A) \cup BND_{\alpha_2}(A) \subseteq [NEG_{\alpha_1}(A)]^c \cap [POS_{\alpha_2}(A)]^c$

6) $BND_{\alpha_1}(A) \cap BND_{\alpha_2}(A)$ =Φ=$[NEG_{\alpha_2}(A)]^c \cap [POS\alpha_1(A)]^c$

Therefore, $BND_{\alpha_1}(A) \cap BND_{\alpha_2}(A) = [NEG_{\alpha_2}(A)]^c \cap [POS\alpha_1(A)]^c$

7) $POS_{\alpha_1}(A) \cup POS_{\alpha_2}(B) = \{a,b,c,d,e\} = POS_{\alpha_1}(A \cup B)$

8) $POS_{\alpha_1}(A) \cap POS_{\alpha_2}(B)$ =Φ = $POS_{\alpha_2}(A \cap B)$

9) $NEG_{\alpha_1}(A) \cup NEG_{\alpha_2}(B)$ =U= $NEG_{\alpha_2}(A \cap B)$

10) $NEG_{\alpha_1}(A) \cap NEG_{\alpha_2}(B)$ =Φ= $NEG_{\alpha_1}(A \cup B)$

11) $BND_{\alpha_1}(A) \cup BND_{\alpha_2}(B)$ =Φ ⊆ U=$[NEG_{\alpha_1}(A \cup B)]^c \cap [POS_{\alpha_2}(A \cap B)]^c$

12) $BND_{\alpha_1}(A) \cap BND_{\alpha_2}(B)$ =Φ=$[NEG_{\alpha_2}(A \cap B)]^c \cap [POS_{\alpha_1}(A \cup B)]^c$

6.7.3 Analysis of the regions of RS-Model using two thresholds

Let X be one of the partitions of U, consider $\{B_1, B_2, \ldots, B_t\}$. The positive, negative and boundary regions of A, the given fuzzy set, with respect to thresholds α and β are given by

1)$POS_{\alpha,\beta}(A) = A_{\alpha,\beta}$

2)$NEG_{\alpha,\beta}(A) = (A^{\alpha,\beta})^C$

3)$BND_{\alpha,\beta}(A) = A^{\alpha,\beta} \cap (A_{\alpha,\beta})^C$

Then the following properties are satisfied.

6.7.3.1 *Property**:***

$POS_{\alpha,\beta}(A \cap B) \supseteq POS_{\alpha,\beta}(A) \cap POS_{\alpha,\beta}(B)$

Proof:

$$POS_{\alpha,\beta}(A)\cap POS_{\alpha,\beta}(B) = A_{\alpha,\beta}\cap B_{\alpha,\beta}$$
$$\subseteq(A\cap B)_{\alpha,\beta}$$
$$=POS_{\alpha,\beta}(A\cap B)$$

6.7.3.2 *Property****:***

$$NEG_{\alpha,\beta}(A\cup B)\supseteq NEG_{\alpha,\beta}(A)\cap NEG_{\alpha,\beta}(B)$$

Proof:

$$NEG_{\alpha,\beta}(A)\cap NEG_{\alpha,\beta}(B) = (A^{\alpha,\beta})^c\cap(B^{\alpha,\beta})^c$$
$$=(A^{\alpha,\beta}\cup B^{\alpha,\beta})^c$$
$$\subseteq[(A\cup B)^{\alpha,\beta}]^c$$
$$=NEG_{\alpha,\beta}(A\cup B)$$

6.7.3.3 *Property****:***

$$(BND_{\alpha,\beta}(A))^c=POS_{\alpha,\beta}(A)\cup NEG_{\alpha,\beta}(A)$$

Proof:

$$(BND_{\alpha,\beta}(A))^c = [A^{\alpha,\beta}\cap(A_{\alpha,\beta})^c]^c$$
$$=(A^{\alpha,\beta})^c\cup A_{\alpha,\beta}$$
$$=NEG_{\alpha,\beta}(A)\cup POS_{\alpha,\beta}(A)$$

Example:

Let U={a,b,c,d,e} be the universe of discourse and X={{a,b},{c},{d,e}} be the partition of U. Suppose A=(0.8,0.4,0.6,0.3,1) and B=(0.5,0.3,0.8,0,0.4) are fuzzy subsets of U. Then D=(0,1)-{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8}. Let $\alpha,\beta\in D$, say, $\alpha=0.45$ and $\beta=0.75$. Also, $A\cup B$=(0.8,0.4,0.8,0.3,1) and $A\cap B$=(0.5,0.3,0.6,0,0.4). Then $A[\alpha,\beta]$={c}, $B[\alpha,\beta]$={a}, $A\cup B[\alpha,\beta]=\Phi$ and $A\cap B[\alpha,\beta]$ ={a,c}.

Therefore, $A_{\alpha,\beta}=\{c\}=A^{\alpha,\beta}$; $B_{\alpha,\beta}=\Phi$; $B^{\alpha,\beta}=\{a,b\}$; $(A\cup B)_{\alpha,\beta}=\Phi=(A\cup B)^{\alpha,\beta}$ and $(A\cap B)_{\alpha,\beta}=\{c\}$; $(A\cap B)^{\alpha,\beta} = \{a,b,c\}$.

Hence, $POS_{\alpha,\beta}(A)=\{c\}$; $NEG_{\alpha,\beta}(A)=\{a,b,d,e\}$; $BND_{\alpha,\beta}(A)= \Phi$; $POS_{\alpha,\beta}(B)= \Phi$; $NEG_{\alpha,\beta}(B)=\{c,d,e\}$; $BND_{\alpha,\beta}(B)=\{a,b\}$; $POS_{\alpha,\beta}(A\cap B)=\{c\}$; $NEG_{\alpha,\beta}(A\cup B) = \{a,b,c,d,e\}$. Then

1)$POS_{\alpha,\beta}(A)\cap POS_{\alpha,\beta}(B)=\{c\}\cap\Phi=\Phi$; $POS_{\alpha,\beta}(A\cap B)=C$

Therefore, $POS_{\alpha,\beta}(A)\cap POS_{\alpha,\beta}\subseteq POS_{\alpha,\beta}(A\cap B)$

2)$NEG_{\alpha,\beta}(A)\cap NEG_{\alpha,\beta}(B)=\{a,b,d,e\}\cap\{c,d,e\}=\{d,e\}$;

$NEG_{\alpha,\beta}(A\cup B)=\{a,b,c,d,e\}$

Therefore, $NEG_{\alpha,\beta}(A)\cap NEG_{\alpha,\beta}(B)\subseteq NEG_{\alpha,\beta}(A\cup B)$

3) $(BND(A))^c=\{a,b,c,d,e\}$

$POS_{\alpha,\beta}(A)\cup NEG_{\alpha,\beta}(A)=\{c\}\cup\{a,b,d,e\}=\{a,b,c,d,e\}$

Hence $(BND(A))^c=POS_{\alpha,\beta}(A)\cup NEG_{\alpha,\beta}(A)$

CHAPTER 7

ROUGH CLASSIFICATION INDUCED BY FUZZY DECISION ATTRIBUTES IN INFORMATION SYSTEM

In this chapter, algorithms are proposed to index the records of the information system, which possesses fuzzy decision attributes. Here, the algorithms are given using lower and upper approximations. This chapter describes the algorithms using single threshold and two thresholds on the fuzzy decision attributes.

7.1 Indices in Databases

For faster random access to the records in a file or database, index structures [1,9] can be used. In association with a particular search key, the index structure will be. There are two basic kinds of indices. They are a) ordered indices and b) hash indices.

The indexing based on a sorted ordering of the values is called ordered indexing. An index-sequential file is ordered sequentially on some search key. There are two types of ordered indices-Dense index and sparse index. For every search key value in the file, the dense index appears whereas in sparse indexing an index record is created only for some of the values.

When the index table is large, the sequential search takes a longer time. In such cases, indices with two or more levels can be used. A form of multilevel index is the B^+ tree. B^+ tree structure is the most widely used index structure. It contains n-1 key values $K_1,K_2,\ldots K_{n-1}$ and n pointers $P_1,P_2,\ldots,P_n$ arranged in the form $P_1K_1P_2K_2\ldots P_{n-1}K_{n-1}P_n$. The search key values are kept in sorted order. The pointer P_i points to either a file record with the bucket of pointers or to a search key value K_i, each of which points to a file record with the search key value K_i. P_n is used to combine the leaf nodes in order of the search key. A B^+ tree is a balanced tree in which the length of every path from the root to a leaf is of same. Each non-leaf node has the children between $\lceil n/2 \rceil$ and n.

The indices of B-tree are similar to indices of B^+ tree but B-tree ignores the redundant storage of search key values.

Hash indices are based on the values being distributed uniformly across a range of buckets. Hash function is useful in determining the bucket, to which a value is assigned. Hashing techniques allow avoiding accessing an index structure. Here, if K is the set of all search key values and the set of all bucket addresses is B, then a function from K to B is the hash function.

In this chapter, the rough indices are developed.

7.2. Rough indices using a threshold

Consider U as the universe of discourse and α be any value in (0,1). Let $X=\{W_1,W_2,\ldots,W_n\}$ be any partition defined on U. For any fuzzy set A define $A[\alpha]=\{x\in U/\ \mu_A(x)>\alpha\}$. The lower and upper approximations A_α and A^α are given by $A_\alpha=\underline{A[\alpha]}$ and $A^\alpha=\overline{A[\alpha]}$ respectively.

The process of indexing the elements of U is illustrated in the below algorithm, by using the lower approximation of the given fuzzy set, A which has the membership values between 0 and 1.

7.2.1. Algorithm for Lower Index of an element

Algorithm lower_index (x,A,α)
//Algorithm to obtain the lower index for element x in universe of discourse
//Algorithm returns the lower index

1. Consider an integer x_index be initialized to 0
2. compute A_α
3. If $x \in A_\alpha$

   while ($x \in A_\alpha$)

     begin

       α=sqrt(α) //square root of α

       x_index=x_index+1

       compute A_α

     end

   else

   while ($x \notin A_\alpha$)

     begin

       α=sqr(α) //square of α

       x_index=x_index-1

       compute A_α

     end
4. return x_index

This algorithm is illustrated with the following example.

Example:

Consider U={a,b,c,d,e,f,g,h} as the universe of discourse with the partition X={{a,e,f},{b,g},{c,h}, {d}}. Let α=0.5. Let the fuzzy set be {(a,0.6),(b,0.4),(c,0.8),(d,0.24), (e,0.44),(f,0.56), (g,0.98), (h,0.77)}. In the first and second iteration, i.e., when α=0.5 and α=0.7071, the elements c and h are in A_α. When α=0.8409, they do not belong to A_α. Hence the lower indices of c and h are 2. Similarly, when α=0.5, the element `a' does not belong to $A_\alpha$ , but when $\alpha$=0.25, `a' belongs to A_α and hence the lower index of `a' is –1. Similarly, the lower index of `d' is –2 and for all other elements the lower index is –1.

Here, the elements of same equivalence classes have the same lower indices, has to be noticed. In the above example, it can also be seen that though the elements `a' and `c' belong to different equivalence classes, they have the same lower index. It relies on upon the decision of α and the fuzzy set taken under consideration. The elements of the same equivalence classes have the same lower indices can be noticed in all the possible cases. Hence, we can follow the given algorithm for lower indexing the equivalence classes, rather than indexing the elements of U.

7.2.2: Algorithm for Lower Index of a set

Algorithm lower_index (K,A,α)

//Algorithm to obtain lower index of K, an equivalence class in universe of discourse

//Algorithm which returns the lower index

1. Consider an integer x_index be initialized to 0
2. compute A_α
3. If $K \subseteq A_\alpha$

   while ($K \subseteq A_\alpha$)

   begin

α=sqrt(α) //square root of α

x_index=x_index+1

compute A_α

end

else

while ($K \not\subset A_\alpha$)

begin

α=sqr(α) //square of α

x_index=x_index-1

compute A_α

end

4. return x_index

The same procedure can be adopted for an upper index by using the upper approximation of A with respect to α. The upper indices of a, b, c, d, e, f, g, h are 1,6,2, -2,1,1,6,2 respectively. Here, the upper indices of the elements of the same equivalence classes are equal. Hence the algorithm for equivalence classes (same as 7.2.2) can also be used. To obtain the upper indices, in algorithm 7.2.2, the definition of lower approximation is to be replaced with the definition of upper approximation.

The above indices have been obtained by using the lower and upper approximations independently. Here an algorithm is given for indexing the values of U by utilizing both the approximations. Consider M be the largest number under consideration such that n+M is constantly positive and n-M is constantly negative for any integer n.

7.2.3: Algorithm for Rough Index of an element

Algorithm index (x,A,α)

//Algorithm to get the index of an element of universe of discourse, x

//Algorithm which returns the index

1. Consider an integer x_index be initialized to 0
2. compute A_α and A^α
3. If $x \in A_\alpha$

```
while (x∈Aα)
  begin
        α=sqrt(α) //square root of α
        x_index=x_index+M+1
        compute Aα
  end
else
        if x∉A^α
        while (x∉A^α)
          begin
                α=sqr(α) //square of α
                x_index=x_index-1-M
                compute A^α
          end
        else
        B=A; β=α
        compute B^β
        while (x∉Aα and x∈B^β)
          begin
                α=sqr(α) // square of α
```

β=sqrt(β) // square root of β

compute A_α, B^β

x_index=x_index+1

end

if $x \in A_\alpha$ then x_index= - x_index

end

4. return x_index

'b' can be indexed by –1 and d can be indexed by –2-M in the given algorithm. Also, different estimations of U can be indexed. These indices are called rough indices. In hashing, the number of collisions is to be minimized. But here, the elements of the same class, i.e., similar elements are indexed with the same number. Hence it is easy to access all similar objects with a single index.

In algorithms 7.2.1 and 7.2.2, it can be observed that the negative and positive indices are obtained by the number of squares and square roots taken for the given α respectively. By using this, it is possible to find the range of membership values to be taken in a fuzzy set to obtain those indices. For example, when α=0.5, if the index of any equivalence class T is –2, then the membership values in the fuzzy set of the elements of T ranges from 0.0625 to 0.25.

7.2.4 Indexing in information system with fuzzy decision attribute using the threshold α

Consider the following knowledge representation of the information system with C={a,b,c,d} and D={E} where E is fuzzy natured.

Table 7.1 Knowledge Representation

	A	**B**	**c**	**D**	**$\mu_E(x_i)$**
x_1	1	0	2	1	0.4
x_2	1	0	2	0	0.8
x_3	1	2	0	0	0.6
x_4	1	2	2	1	0
x_5	2	1	0	0	1
x_6	2	1	1	0	0.6
x_7	2	1	2	1	0.3

Consider 'c' as the index key. As x_1,x_2,x_4,x_7 have the values 2; x_3,x_5 have the values 0 and x_6 has the value 1. Hence, the partition on U can be defined as $\{\{x_1,x_2,x_4,x_7\},\{x_3,x_5\},\{x_6\}\}$.

Let α=0.5. Here, $A[\alpha]=\{x_2,x_3,x_5,x_6\}$. Hence, $A_\alpha=\{x_3,x_5,x_6\}$ and $A^\alpha=\{x_1,x_2,x_3,x_4,x_5,x_6, x_7\}$. By using the algorithm 7.2.3, the records can be indexed.

7.2.5 Indexing using reducts

In the section 7.2.4, the index key was chosen based on the choice of the user. To make the indexing be efficient, the index key is to be chosen using the appropriate reduct, which was discussed in the previous chapter.

7.3 Rough indices by two thresholds

Consider the universe of discourse U and α be any value in (0,1). Consider $X=\{W_1,W_2,\ldots,W_n\}$ be any partition defined on U. A define $A[\alpha,\beta]=\{x\in U/\ \alpha<\mu_A(x)\leq\beta\}$, for any fuzzy set A. The lower and upper approximations $A_{\alpha,\beta}$ and $A^{\alpha,\beta}$ are given by $A_{\alpha,\beta}=\underline{A[\alpha,\beta]}$ and $A^{\alpha,\beta}=\overline{A[\alpha,\beta]}$ respectively.

By utilizing the lower approximation of the given fuzzy set A, the below algorithm outlines the technique of indexing the elements of U. It has the membership values between 0 and 1.

7.3.1: Algorithm for Lower Index of an element

Algorithm lower_index (x,A,α,β)

/Algorithm to obtain the lower index for element x in universe of discourse

//Algorithm which returns the lower index

1. Consider an integer x_index be initialized to 0
2. compute $A_{\alpha,\beta}$
3. If $x\in A_{\alpha,\beta}$

   while ($x\in A_{\alpha,\beta}$)

   begin

```
α=sqrt(α) //square root of α
β=sqr(β) //square of β
x_index=x_index+1
compute A_{α,β}
 end
else
while (x∉A_{α,β})
 begin
  α=sqr(α) //square of α
  β=sqrt(β) //square root of β
  x_index=x_index-1
  compute A_{α,β}
 end
```

4. return x_index

This algorithm is illustrated by the following example.

7.3.1.1. *Example:*

Consider U={a,b,c,d,e,f,g,h} as the universe of discourse with the partition X={{a,e,f},{b,g},{c,h}, {d}}. Let α=0.5 and β=0.85. Let the fuzzy set be {(a,0.6),(b,0.4),(c,0.8),(d,0.24), (e,0.44),(f,0.56), (g,0.98), (h,0.77)}. It is noted that during the first iteration, i.e., when α=0.5 and β=0.85, the elements c and h are in $A_{\alpha,\beta}$. When α=0.7071 and β=0.7225, they do not belong to $A_{\alpha,\beta}$. Hence the lower indices of c and h are 1. Similarly, when α=0.5 and β=0.85, the element `a' does not belong to $A_{\alpha,\beta}$, but when $\alpha$=0.25 and $\beta$=0.9219, `a' belongs to $A_{\alpha,\beta}$ and hence the lower index of `a' is –1. Similarly, the lower indices of other elements can be found.

The elements of same equivalence classes have the same lower indices has to be noticed here. In the above example, it can also be seen that though the elements `a' and `c' belong to different equivalence classes, they have the same lower index. It depends on the choice of α and β and the fuzzy set taken under consideration. The elements of the same equivalence classes have the same lower indices can be noticed in all the possible cases. Consequently, we can follow the given algorithm for lower indexing the equivalence classes, rather than indexing the elements of U.

7.3.2: Algorithm for Lower Index of a Set

Algorithm lower_index (K,A,α,β)

//Algorithm to obtain the lower index of K, an equivalence class in universe of discourse

//Algorithm which returns the lower index

1. Consider an integer x_index be initialized to 0
2. compute $A_{\alpha,\beta}$
3. If $K \subseteq A_{\alpha,\beta}$

   while ($K \subseteq A_{\alpha,\beta}$)

   begin

   α=sqrt(α) //square root of α

   β=sqr(β) //square of β

   x_index=x_index+1

   compute $A_{\alpha,\beta}$

   end

   else

   while ($K \not\subseteq A_{\alpha,\beta}$)

   begin

   α=sqr(α) //square of α

   β=sqrt(β) //square root of β

   x_index=x_index-1

compute $A_{\alpha,\beta}$

end

4. return x_index

The same procedure can be adapted for an upper index by using the upper approximation of A with respect to α and β. Here, the upper indices of the elements of the same equivalence classes are equal. Hence the algorithm for equivalence classes (same as 7.3.2) can also be used. To obtain the upper indices, in algorithm 7.3.2, the definition of lower approximation is to be replaced with the definition of upper approximation.

The above indices have been obtained by using the lower and upper approximations independently. Here an algorithm is given for indexing the values of U by using both the approximations [16]. Consider M denotes the biggest number under consideration such that n+M is constantly positive and n-M is constantly negative for any integer n.

7.3.3: Algorithm for Rough index of an element

Algorithm index (x,A,α,β)

//Algorithm to obtain the index of x an element of universe of discourse

//Algorithm which returns the index

1. Consider an integer x_index be initialized to 0
2. Compute $A_{\alpha,\beta}$ and $A^{\alpha,\beta}$
3. If $x \in A_{\alpha,\beta}$

while ($x \in A_{\alpha,\beta}$)

begin

α=sqrt(α) //square root of α

β=sqr(β) //square of β

```
            x_index=x_index+M+1
            compute A_{α,β}
        end
    else
            if x∉A^{α,β}
            while (x∉A^{α,β})
              begin
                    α=sqr(α) //square of α
                    β=sqrt(β) //square root of β
                    x_index=x_index-1-M
                    compute A^{α,β}
              end
            else
            B=A; χ=α; δ=β
            compute B^{χ,δ}
            while (x∉A_{α,β} and x∈B^{χ,δ})
             begin
                    α=sqr(α) // square of α
                    β=sqrt(β) //square root of β
                    χ=sqrt(χ) // square root of χ
                    δ=sqr(δ) //square of δ
                    compute A_{α,β},B^{χ,δ}
                    x_index=x_index+1
             end
            if x∈A_{α,β} then x_index= - x_index
    end
4.  return x_index
```

`b' can be indexed by –2-M and 'a' can be indexed by –2 by using the above algorithm. Similarly, the other values of U can be indexed. These indices are called rough indices. In hashing, the number of collisions is to be minimized. But here, the elements of the same class, i.e., similar elements are indexed with the same number. Hence, it is easy to access all similar objects with a single index.

7.3.4 Indexing in Information System with fuzzy decision attribute using two thresholds

Consider the following knowledge representation of the information system with C={a,b,c,d} and D={E} where E is fuzzy natured.

Table 7.2 Knowledge Representation

	A	B	C	D	$\mu_E(x_i)$
x_1	1	0	2	1	0.4
x_2	1	0	2	0	0.8
x_3	1	2	0	0	0.6
x_4	1	2	2	1	0
x_5	2	1	0	0	1
x_6	2	1	1	0	0.6
x_7	2	1	2	1	0.3

Let α=0.5 and β=0.9. Here, $A[\alpha,\beta]=\{x_2,x_3,x_5,x_6\}$. Hence, $A_{\alpha,\beta}=\{x_3,x_5,x_6\}$ and $A^{\alpha}=\{x_1,x_2,x_3,x_4,x_5,x_6, x_7\}$. By using the algorithm 7.3.3, the records can be indexed.

In section 7.3.4, the index key was chosen based on the choice of the user. To make the indexing be efficient, for the given α from R domain, the index key is to be chosen using the appropriate reduct.

CHAPTER 8

PROBABILISTIC ROUGH CLASSIFICATION IN INFORMATION SYSTEMS WITH FUZZY DECISION ATTRIBUTES

As discussed earlier, the limitation of Pawlak's approximation is that it does not quantify the level of importance of the basic granules. Recently, Y.Y.Yao discussed Probabilistic Rough Set Model, which specified how basic granules could be quantified appropriately, through which we have developed an indexing algorithm on information system with fuzzy decision attributes.

8.1. Decision-Theoretic and Probabilistic Rough Sets

Rough Sets theory defines two-way approximations for a given input namely lower and upper approximations. The equivalence class of any x∈U is defined to be $[x]=\{y\in U / xEy\}$ for a given finite universe of discourse U and an equivalence relation E. The family of equivalence classes $U/E=\{[x]_E \mid x\in U\}$ is a partition of the universe U. Pawlak has defined the lower approximation as $\underline{apr}_E(C)=\{x\in U/[x]_E\subseteq C\}$ and upper approximation as $\overline{apr}_E(C)=\{x\in U/[x]_E\cap C\neq\Phi\}$, for a given concept C.

Three disjoint regions for a given concept C, namely positive, negative and boundary regions are defined as follows:

Positive Region: $POS_E(C)=\{x\in U/[x]_E\subseteq C\}$ **(8.1)**

Boundary : $BND_E(C)=\{x\in U/[x]_E \cap C \neq \Phi \wedge [x]_E \not\subseteq C\}$ **(8.2)**

Negative region: $NEG_E(C)=\{x\in U/[x]_E \cap \in \quad \Phi\}$ **(8.3)**

As Pawlak's model is restrictive, many researchers concentrated on generalizing this approach against parameterized rough set model, probabilistic rough set model and generalized the rough set model.

Pawlak and Skowron in 1994, have stated rough membership function in consideration of overlap between equivalence classes and a concept C to be approximated and is observed as the conditional probability of an object belongs to C provided that the object is in [x] (for simplicity, we denote $[x]_E$ with [x]) which is given as $\Pr\left(C/[x]\right)=\frac{|C\cap[x]|}{\|[x]\|}$

The positive, boundary and negative regions are stated as follows by using the definition quoted above:

$$POS(C)=\left\{x\in U \;/\; \Pr\left(C/[x]\right)=1\right\}$$ **(8.4)**

$$BND(C)=\left\{x\in U \;/\; 0<\Pr\left(C/[x]\right)<1\right\}$$ **(8.5)**

$$NEG(C)=\left\{x\in U \;/\; \Pr\left(C/[x]\right)=0\right\}$$ **(8.6)**

By generalizing the above-said definitions, in 2009, Greco et.al exchange views on the parameterized rough set model [23]. To define the probabilistic regions, two thresholds namely α and β are used in this model and the positive, boundary and negative regions are modified as follows:

$$POS_{(\alpha,\beta)}(C)=\left\{x\in U \;/\; \Pr\left(C/[x]\right)\geq\alpha\right\}$$ **(8.7)**

$$BND_{(\alpha,\beta)}(C)=\left\{x\in U \;/\; \beta<\Pr\left(C/[x]\right)<\alpha\right\}$$ **(8.8)**

$$NEG_{(\alpha,\beta)}(C) = \left\{ x \in U \ / \Pr\left(C/_{[x]} \right) \leq \beta \right\} \quad \textbf{(8.9)}$$

For any object x in U, these Probabilistic regions will prompt to three-way decisions namely deferment, acceptance, and rejection respectively. In any case, be that as it may, in a few cases, it is anything but difficult for a given concept C, to process the likelihood of the presence of a category [x] using $\Pr\left([x]/_{C} \right) = \frac{|[x] \cap C|}{|C|}$

However, by Baye's Theorem, the Positive, Boundary and Negative Regions are given by

$$POS^{B}_{(\alpha',\beta')}(C) = \left\{ x \in U \ / \log \frac{\Pr([x]/C)}{\Pr([x]/C^{c})} \geq \alpha' \right\} \quad \textbf{(8.10)}$$

$$BND^{B}_{(\alpha',\beta')}(C) = \left\{ x \in U \ / \beta' < \log \frac{\Pr([x]/C)}{\Pr([x]/C^{c})} < \alpha' \right\} \quad \textbf{(8.11)}$$

$$NEG^{B}_{(\alpha',\beta')}(C) = \left\{ x \in U \ / \log \frac{\Pr([x]/C)}{\Pr([x]/C^{c})} \leq \beta' \right\} \quad \textbf{(8.12)}$$

$$where \quad \alpha' = \log \frac{\Pr(C^{c})}{\Pr(C)} + \log \frac{\alpha}{1-\alpha}$$

$$and \quad \beta' = \log \frac{\Pr(C^{c})}{\Pr(C)} + \log \frac{\beta}{1-\beta}$$

8.2. Naïve Bayesian Probabilistic Rough Sets Model for a Fuzzy Concept

The same threshold α has been utilized in the above sections, for different purposes. In this chapter, we replace the threshold α to obtain a Strong Cut on fuzzy sets with δ to achieve the homogeneity.

Thus, the probabilistic positive, boundary and negative regions for a given fuzzy concept F with the threshold δ are individually characterized on the approximation space U/E as

$$POS_{\delta}(F) = \left\{ x \in U / \Pr\left({F[\delta]}/{[x]} \right) = 1 \right\} \tag{8.13}$$

$$BND_{\delta}(F) = \left\{ x \in U / 0 < \Pr\left({F[\delta]}/{[x]} \right) < 1 \right\} \tag{8.14}$$

$$NEG_{\delta}(F) = \left\{ x \in U / \Pr\left({F[\delta]}/{[x]} \right) = 0 \right\} \tag{8.15}$$

The regions of the parameterized rough sets model for the provided parameters α and β, are given by

$$POS_{(\alpha,\beta,\delta)}(F) = \left\{ x \in U / \Pr\left({F[\delta]}/{[x]} \right) \geq \alpha \right\} \tag{8.16}$$

$$BND_{(\alpha,\beta,\delta)}(F) = \left\{ x \in U / \beta < \Pr\left({F[\delta]}/{[x]} \right) < \alpha \right\} \tag{8.17}$$

$$NEG_{(\alpha,\beta,\delta)}(F) = \left\{ x \in U / \Pr\left({F[\delta]}/{[x]} \right) \leq \beta \right\} \tag{8.18}$$

and the Naïve Bayesian Rough Sets Model regions are given by

$$POS^{B}_{(\alpha',\beta',\delta)}(F) = \left\{ x \in U / \log \frac{\Pr\left([x]/F[\delta]\right)}{\Pr\left([x]/(F[\delta])^{C}\right)} \geq \alpha' \right\} \tag{8.19}$$

$$BND^{B}_{(\alpha',\beta',\delta)}(F) = \left\{ x \in U / \beta' < \log \frac{\Pr\left([x]/F[\delta]\right)}{\Pr\left([x]/(F[\delta])^{C}\right)} < \alpha' \right\} \tag{8.20}$$

$$NEG^{B}_{(\alpha',\beta',\delta)}(F) = \left\{ x \in U / \log \frac{\Pr\left([x]/F[\delta]\right)}{\Pr\left([x]/(F[\delta])^{C}\right)} \leq \beta' \right\} \quad \textit{where} \quad \alpha' = \log \frac{\Pr(C^{c})}{\Pr(C)} + \log \frac{\alpha}{1-\alpha}$$

$$\textit{and} \quad \beta' = \log \frac{\Pr(C^{c})}{\Pr(C)} + \log \frac{\beta}{1-\beta} \tag{8.21}$$

8.3. Rough Indices

Here, we modify the rough indices algorithm [discussed earlier] for three-way approach on rough sets as follows:

Algorithm Three_Way_ rough index (x,A,δ)

//Algorithm which returns the Three_Way_rough index of x

1. Consider and integer x_index initialized to 0
2. Choose the equivalence class K containing x.

        If $\mu_A(y)=0$ for all $y \in K$
        begin
            x_index=-M
            goto 6
        end
    end
3. If $\mu_A(x)=1$

    begin
        If $\mu_A(y)=1$ for all $y \in K$
        begin
            x_index=M
            goto 6
        end
    end
4. compute $POS_\delta(A)$, $BND_\delta(A)$ and $NEG_\delta(A)$
5. If $x \in POS_\delta(A)$

    begin
    x_index=M
    while ($x \in POS_\delta(A)$)
        begin

```
          α=dil(δ)          //dilation of δ
          x_index=x_index+1
          compute POS_δ(A)
   end
end
else
          if x∈ NEG_δ(A)
          begin
          x_index=-M
          while (x∈ NEG_δ(A))
             begin
                δ=con(δ)//concentration of δ
               x_index=x_index-1
                compute NEG_δ(A)
              end
          end
          else
          Let γ=δ
          compute NEG_γ(A)
          while (x∉(POS_δ(A)∪NEG_γ(A)))
            begin
              δ=con(δ) // concentration of δ
              γ=dil(γ)      // dilation of γ
              compute POS_δ(A)∪NEG_γ(A)
              x_index=x_index+1
            end
          if x∈ POS_δ(A) then
            x_index= - x_index
end
```

6. return x_index

Now, we parameterize the algorithm using parameters α and β.

Algorithm Naïve Bayesian_ rough index (x,A,α,β,δ)

//Algorithm returns Naïve Bayesian_rough index of x

1. Consider an integer x_index initialized to 0
2. Choose the equivalence class K containing x.

   If $\mu_A(y)=0$ for all $y\in K$

   begin

   x_index=-M

   goto 6

   end

   end
3. If $\mu_A(x)=1$

   begin

   If $\mu_A(y)=1$ for all $y\in K$

   begin

   x_index=M

   goto 6

   end

   end
4. compute $POS^B_{(\alpha',\beta',\delta)}(A)$, $BND^B_{(\alpha',\beta',\delta)}(A)$ and $NEG^B_{(\alpha',\beta',\delta)}(A)$
5. If $x\in POS^B_{(\alpha',\beta',\delta)}(A)$

   begin

   x_index=M

   while ($x\in POS^B_{(\alpha',\beta',\delta)}(A)$)

   begin

   α=dil(δ) //dilation of δ

```
                x_index=x_index+1
                compute POS^B_(α',β',δ)(A)
            end
        end
else
                if x∈ NEG^B_(α',β',δ)(A)
                begin
                x_index=-M
                while (x∈ NEG^B_(α',β',δ)(A))
                  begin
                    δ=con(δ)//concentration of δ
                    x_index=x_index-1
                    compute NEG^B_(α',β',δ)(A)
                   end
                end
                else
                Let γ=δ
                compute NEG_γ(A)
                while
                    (x∉(POS^B_(α',β',δ)(A)∪NEG^B_(α',β',γ)(A)))
                  begin
                   δ=con(δ) // concentration of δ
                   γ=dil(γ)     // dilation of γ
                   compute
                     POS^B_(α',β',δ)(A)∪NEG^B_(α',β',γ)(A)
                   x_index=x_index+1
                  end

                if x∈ POS^B_(α',β',δ) (A) then
```

x_index= - x_index

end

6. return x_index

8.4. Naïve Bayesian Indexing in Information System with Fuzzy Decision Attribute

Any information system is given by T=(U, A, C, D) as per the description of Z.Pawlak, where U is the universe of discourse, A is a set of primitive attributes, C and D are the subsets of A which are termed as condition and decision features respectively [for few of the information systems, C and D may not exist].

Consider an information system with conditional attributes $C=\{a_1,a_2,\ldots,a_n\}$ and decision attributes $\{d_1,d_2,\ldots,d_s\}$ with the records $U=\{x_1,x_2,\ldots,x_m\}$. For any index key 'a' in C, the indiscernibility relation is given by $x_i \approx_{a_k} x_j$ (read as x_i is related to x_j with respect to a_k) if and only if $a_k(x_i)=a_k(x_j)$. It is clearly seen that this indiscernibility relation partitions U, the universe of discourse. The selection process of the appropriate minimal attributes [reducts] for effectiveness is not discussed in this chapter.

Consider the decision table with C={a,b,c,d} and D={E}.

Table 8.1 Decision Table

	A	b	C	D	E
x_1	1	0	2	1	1
x_2	1	0	2	0	1

x_3	1	2	0	0	2
x_4	1	2	2	1	0
x_5	2	1	0	0	2
x_6	2	1	1	0	2
x_7	2	1	2	1	1

Consider ‘c’ as the index key. As x_1,x_2,x_4,x_7 have the values 2; x_3,x_5 have the values 0 and x_6 has the value 1. Thus, the partition on U with respect to c can be characterized as $\{\{x_1,x_2,x_4,x_7\},\{x_3,x_5\},\{x_6\}\}$.

However, we can discover several information systems with fuzzy decision attributes in real-time systems and henceforth the extent of the algorithms discussed above would be applicable to such information systems. As discussed in the previous section, the Naïve Bayesian rough indexing of the data can be derived from the fuzzy decision attribute.

For instance, consider the knowledge representation of the information system with C={a,b,c,d} and D={E} where E is fuzzy natured.

Table 8.2 Knowledge Representation

	A	B	C	D	$\mu_E(x_i)$
x_1	1	0	2	1	0.45
x_2	1	0	2	0	0.7

x_3	1	2	0	0	0.65
x_4	1	2	2	1	0.1
x_5	2	1	0	0	0.91
x_6	2	1	1	0	0.6

On considering ‘c’ as the index key, the partition acquired is $\{\{x_1,x_2,x_4,x_7\}, \{x_3,x_5\},\{x_6\}\}$. Let δ=0.5. Here, $E[\delta] = \{x_2,x_3,x_5,x_6\}$. For a given α and β, the Naïve Bayesian indexing algorithm would be executed further.

CHAPTER 9

INTUITIONISTIC FUZZINESS AND CONCEPTS OF THRESHOLDS UNDER ROUGHNESS

9.1 Intuitionistic Fuzzy Sets

According to the concept of fuzzy sets, the non-membership function gives a value that is equivalent to the variation between the membership value and one. However, this cannot be used in all applications, because some of them defy this nonmembership rule. These particular applications prompted Atanassov to develop the concept of the intuitionistic fuzzy set [3, 21]. The path he followed allows the dimensional view of fuzzy sets in terms of membership grade and non-membership grade.

9.1.1 Definition:

Let U $=\{x_1,x_2,\ldots x_n\}$ be any universe of discourse and A be any intuitionistic fuzzy subset of U. If μ_A and γ_A be the membership and non-membership functions defined on the universe of discourse U to [0,1] with $0\le\mu_A(x)+\gamma_A(x)\le1$ for every x in U, Then the intuitionistic fuzzy subset A of U is denoted by

$$A=\{\frac{(\mu_A(x_1),\gamma_A(x_1))}{x_1}+\frac{(\mu_A(x_2),\gamma_A(x_2))}{x_2}+\ldots+\frac{(\mu_A(x_n),\gamma_A(x_n))}{x_n}+\ldots\}$$

9.1.2 Operations

Consider A and B as any two intuitionistic fuzzy sets. Then the accompanying operations on A and B are characterized.

1) Union:

$$A \cup B = \{\frac{((\max(\mu_A(x_1), \mu_B(x_1)), \min(\gamma_A(x_1), \gamma_B(x_1)))}{x_1} + \frac{((\max(\mu_A(x_2), \mu_B(x_2)), (\min(\gamma_A(x_2), \gamma_B(x_2)))}{x_2} + \ldots\}$$

2) Intersection:

$$A \cap B = \{\frac{((\min(\mu_A(x_1), \mu_B(x_1)), \max(\gamma_A(x_1), \gamma_B(x_1)))}{x_1} + \frac{((\min(\mu_A(x_2), \mu_B(x_2)), (\max(\gamma_A(x_2), \gamma_B(x_2)))}{x_2} + \ldots\}$$

3) Complement:

$$\overline{A} = \{\frac{(\gamma_A(x_1), \mu_A(x_1))}{x_1} + \frac{(\gamma_A(x_2), \mu_A(x_2))}{x_2} + \ldots\}$$

4) Addition:

$$A+B = \{\frac{(\mu_A(x_1) + \mu_B(x_1) - \mu_A(x_1) \cdot \mu_B(x_1), \gamma_A(x_1) \cdot \gamma_B(x_1))}{x_1} + \ldots\}$$

5) Product:

$$A \bullet B = \{\frac{(\mu_A(x_1) \cdot \mu_B(x_1), \gamma_A(x_1) + \gamma_B(x_1) - \gamma_A(x_1). \cdot \gamma_B(x_1))}{x_1} + \ldots\}$$

9.1.2.1 *Remarks*

(i) The operations intersection ($\cap$) and union ($\cup$) satisfy commutativity, associativity, left and right distributivity among themselves, idempotency and De Morgan’s laws.

(ii) The operations **+** and $\cdot$ are commutative, associative and satisfy a law which is similar to De Morgan. Also, these operations satisfy the left and right distributivity with respect to the operations $\cap$ and $\cup$.

9.1.3 Subset

Let A and B be any two intuitionistic fuzzy subsets of U. If $\mu_A(x) \leq \mu_B(x)$ and $\gamma_A(x) \geq \gamma_B(x)$ for all $x \in U$, then it is said that A is subset of B. If A is subset of B and B is subset of A, then A and B are said to be equal.

9.2 (α,β)- cut of intuitionistic fuzzy set

Suppose U is any universe of discourse and A is an intuitionistic fuzzy set, then the (α,β)-cut of A, denoted by $A_{\alpha,\beta}$ is a crisp set which can be defined as $A_{\alpha,\beta} = \{x \in U: \mu_A(x) \geq \alpha , \gamma_A(x) \leq \beta\}$.

9.2.1 Properties:

Consider A and B as two intuitionistic fuzzy sets of a universe U. Then the following properties are satisfied

(i) $A_{\alpha,\beta} \subseteq A_{\delta,\theta}$ if $\alpha \geq \delta$ and $\beta \leq \theta$

(ii) $A_{1-\beta,\beta} \subseteq A_{\alpha,\beta} \subseteq A_{\alpha,1-\alpha}$

(iii) $A \subseteq B$ implies $A_{\alpha,\beta} \subseteq B_{\alpha,\beta}$

(iv) $(A \cap B)_{\alpha,\beta} = A_{\alpha,\beta} \cap B_{\alpha,\beta}$

(v) $(A \cup B)_{\alpha,\beta} \supseteq A_{\alpha,\beta} \cup B_{\alpha,\beta}$ equality holds if $\alpha+\beta=1$

(vi) $(\cap A_i)_{\alpha,\beta} = \cap(A_i)_{\alpha,\beta}$

(vii) $A_{0,1} = U$

9.3 Intuitionistic Rough Fuzzy Sets

Consider the finite universe of discourse $U=\{x_1,x_2,\ldots x_n\}$ and $W=\{X_1,X_2,\ldots,X_t\}$ be a partition of U. For any intuitionistic fuzzy subset A of U, define

$$\mu_{\underline{A}}(X_i) = \inf\{\mu_A(x_j) : x_j \in X_i\}$$

$$\gamma_{\underline{A}}(X_i) = \sup\{\gamma_A(x_j) : x_j \in X_i\}$$

$$\mu_{\overline{A}}(X_i) = \sup\{\mu_A(x_j) : x_j \in X_i\}$$

$$\gamma_{\overline{A}}(X_i) = \inf\{\gamma_A(x_j) : x_j \in X_i\}$$

The lower and upper approximations can be defined as:

$$\underline{A} = [(\mu_{\underline{A}}(X_1), \gamma_{\underline{A}}(X_1)), \ldots, (\mu_{\underline{A}}(X_t), \gamma_{\underline{A}}(X_t))]$$

$$\overline{A} = [(\mu_{\overline{A}}(X_1), \gamma_{\overline{A}}(X_1)), \ldots, (\mu_{\overline{A}}(X_t), \gamma_{\overline{A}}(X_t))]$$ respectively.

The matrix $\begin{bmatrix} \mu_{\underline{A}}(X_i) & \gamma_{\underline{A}}(X_i) \\ \mu_{\overline{A}}(X_i) & \gamma_{\overline{A}}(X_i) \end{bmatrix}$ is termed as the intuitionistic fuzzy matrix for each X_i.

9.4 Generalized Intuitionistic Rough Fuzzy Sets

Four kinds of generalizations were introduced on IRFS namely,

a) **MCNC**: Certainty approximations on both Membership and Non-Membership grades

b) **MCNP**: Certainty approximation on Membership and Possibility approximation on Non-Membership grades

   a. **MPNC**: Possibility approximation on Membership grades and certainty approximation on Non-Membership grades

c) **MPNP**: Possibility approximations on both Membership and Non-Membership grades.

Consider a finite universe of discourse U={a_1,a_2,…a_t}and let R be a relation defined on U. Let U/R={X_1,X_2,…X_n}. Let A be any intuitionistic fuzzy subset of U.

a) MCNC Approximations

The functions $\alpha_i,\beta_{ij},\rho_i$ and δ_{ij} for MCNC approximations are defined as

$$\alpha_i = \inf\left\{\mu_A(a_j) : a_j \in X_i\right\} \quad \text{and} \quad \beta_{ij} = \begin{cases} \alpha_i & if \quad a_j \in X_i \\ 0 & otherwise \end{cases}$$

$$\rho_i = \sup\left\{\gamma_A(a_j) : a_j \in X_i\right\} \text{and} \quad \delta_{ij} = \begin{cases} \rho_i & if \quad a_j \in X_i \\ 1 & otherwise \end{cases}$$

Define $M_i = \max(\beta_{i1},\beta_{i2},.....,\beta_{it})$ and $NM_i = \min(\delta_{i1},\delta_{i2},.....,\delta_{it})$

The MCNC approximation of A is defined as

$$MCNC_R(A) = \left\{\frac{(M_j, NM_j)}{X_j}\right\}_{j=1}^{n}$$

b) MCNP Approximations

The functions $\alpha_i,\beta_{ij},\rho_i$ and δ_{ij} for MCNP approximations are defined as

$$\alpha_i = \inf\left\{\mu_A(a_j) : a_j \in X_i\right\} \quad \text{and} \quad \beta_{ij} = \begin{cases} \alpha_i & if \quad a_j \in X_i \\ 0 & otherwise \end{cases}$$

$$\rho_i = \inf\left\{\gamma_A(a_j) : a_j \in X_i\right\} \quad \text{and} \quad \delta_{ij} = \begin{cases} \rho_i & if \quad a_j \in X_i \\ 1 & otherwise \end{cases}$$

Define $M_i = \max(\beta_{i1},\beta_{i2},.....,\beta_{it})$ and $NM_i = \min(\delta_{i1},\delta_{i2},.....,\delta_{it})$

The MCNP approximation of A is defined as $MCNP_R(A) = \left\{\frac{(M_j, NM_j)}{X_j}\right\}_{j=1}^{n}$

c) MPNC Approximations

The functions $\alpha_i,\beta_{ij},\rho_i$ and δ_{ij} for MPNC approximations are defined as

$$\alpha_i = \sup\left\{\mu_A(a_j) : a_j \in X_i\right\} \quad \text{and} \quad \beta_{ij} = \begin{cases} \alpha_i & if \quad a_j \in X_i \\ 0 & otherwise \end{cases}$$

$$\rho_i = \sup\left\{\gamma_A(a_j) : a_j \in X_i\right\} \quad \text{and} \quad \delta_{ij} = \begin{cases} \rho_i & if \quad a_j \in X_i \\ 1 & otherwise \end{cases}$$

Define $NM_i = \min\left(\delta_{i1}, \delta_{i2}, ...,\delta_{it}\right)$ and

$$M_i = \begin{cases} \max\limits_{1\le j\le t}\left\{\beta_{ij}\right\} & if \quad \max\limits_{1\le j\le t}\left\{\beta_{ij}\right\} + NM_i < 1 \\ 1 - NM_i & otherwise \end{cases}$$

The MPNC approximation of A is defined as $MPNC_R(A) = \left\{\dfrac{(M_j, NM_j)}{X_j}\right\}_{j=1}^{n}$

d) **MPNP Approximations**

For MPNP approximations, the functions α_i,β_{ij},ρ_i and δ_{ij} are defined as

$$\alpha_i = \sup\left\{\mu_A(a_j) : a_j \in X_i\right\} \quad \text{and} \quad \beta_{ij} = \begin{cases} \alpha_i & if \quad a_j \in X_i \\ 0 & otherwise \end{cases}$$

$$\rho_i = \inf\left\{\gamma_A(a_j) : a_j \in X_i\right\} \quad \text{and} \quad \delta_{ij} = \begin{cases} \rho_i & if \quad a_j \in X_i \\ 1 & otherwise \end{cases}$$

Define $M_i = \max\left(\beta_{i1}, \beta_{i2}, ...,\beta_{it}\right)$ and $NM_i = \min\left(\delta_{i1}, \delta_{i2}, ...,\delta_{it}\right)$

The MPNP approximation of A is defined as $MPNP_R(A) = \left\{\dfrac{(M_j, NM_j)}{X_j}\right\}_{j=1}^{n}$

Similar to fuzzy environment, the connectives are defined under Intuitionistic fuzziness through rough sets as follows:

9.5. Rough Connectives of Intuitionistic Fuzzy Predicates

For the given intuitionistic fuzzy predicate P, denote $\{\mu_{P(x)}, \gamma_{P(x)}\}$ as the doublet consisting the grades of membership μ_x and non-membership γ_x of P(x). Then, the m-negation [m-neg] and nm-negation [nm-neg] of P(x) are respectively given by $\{1-\mu_{P(x)}, \mu_{P(x)}\}$ and $\{1-\gamma_{P(x)}, \gamma_{P(x)}\}$.

Consider the collection of intuitionistic fuzzy predicates $\{P_1,P_2,\ldots,P_k\}$ and the arguments $\{x_1,x_2,\ldots,x_n\}$.

Let X be any partition defined on the collection of all arguments using some equivalence relation. Then P_i can be denoted as

$P_i = \left(\{\mu_{P_i(x_1)}, \gamma_{P_i(x_1)}\}, \{\mu_{P_i(x_2)}, \gamma_{P_i(x_2)}\}, \ldots, \{\mu_{P_i(x_n)}, \gamma_{P_i(x_n)}\}\right)$ Define the set

$M_{ij} = \{s : s = \mu_{P_j(x_i)} \text{ or } s = \gamma_{P_j(x_i)} \text{ or } s = 1-\mu_{P_j(x_i)} \text{ or } s = 1-\gamma_{P_j(x_i)}\}$ Define $D = (0,1) - \bigcup_{i=1}^{n}\bigcup_{j=1}^{k} M_{ij}$ and let $\alpha \in D$.

Define $P[\alpha]=\{x: \mu_{P(x)} > \alpha \text{ and } \gamma_{P(x)} < 1-\alpha\}$.

The lower and upper rough approximations are defined by

$P_\alpha = \cup\{[x] \in X : [x] \subseteq P[\alpha]\}$ and

$P^\alpha = \cup\{[x] \in X : [x] \cap P[\alpha] \neq \Phi\}$ respectively.

If $x \in P_\alpha$ then define $P_\alpha(x)$ is true otherwise it is false. If $x \in P^\alpha$, then define $P^\alpha(x)$ is true otherwise it is false. As $P_\alpha \subseteq P^\alpha$, if $P_\alpha(x)$ is true then $P^\alpha(x)$ is true. Thus, for each intuitionistic fuzzy predicate P, the lower and upper predicates, which are crisp, can be defined with respect to α.

Here, we introduce two types of Complements based on membership and non-membership grades.

$P_i^{m(c)} = m - neg(P_i)$; $P_i^{nm(c)} = nm - neg(P_i)$

$(P_i[\alpha])^{m(c)} = \{x : \mu_{P(x)} < \alpha\}$ and $(P_i[\alpha])^{nm(c)} = \{x : \gamma_{P(x)} > 1 - \alpha\}$

9.5.1. Rough Connectives on Intuitionistic Fuzzy Predicates

In this section, the connectives are introduced similar to the connectives used in the predicate calculus.

Definition:

For the given intuitionistic fuzzy predicates $P_i(x)$ and $P_j(y)$, the lower and upper conjunctions $\underset{\alpha}{\wedge}$ and $\overset{\alpha}{\wedge}$ are defined as

$P_i(x) \underset{\alpha}{\wedge} P_j(y) = P_{i,\alpha}(x) \wedge P_{j,\alpha}(y)$ and

$P_i(x) \overset{\alpha}{\wedge} P_j(y) = P_i^{\alpha}(x) \wedge P_j^{\alpha}(y)$ respectively.

Here, $(\underset{\alpha}{\wedge}, \overset{\alpha}{\wedge})$ is called the **rough conjunction**.

Example:

Let the universe of discourse having U={a,b,c,d,e,f} with the partition Ψ={{a,c,e},{b,f},{d}}. Consider the intuitionistic fuzzy predicates P and Q defined on U which are given by

P=((0.2,0.79),(0.6,0.39),(0.5,0.49),(0.4,0.59),(0.3,0.68),(0.7,0.28)) and

Q=((0.4,0.58),(0.6,,0.38), (0.3,0.69),(0.6,0.39),(0.5,0.47),(0.8,0.19)).

Let α=0.45. Then P[α]={b,c,f} and Q[α]={b,d,e,f}. Hence, P_{α}={b,f}, P^{α}={a,b,c,e,f}, Q_{α}={b,d,f} and Q^{α}=Ψ

x_1	x_2	$P(x_1)\underset{\alpha}{\wedge}Q(x_2)$	$P(x_1)\overset{\alpha}{\wedge}Q(x_2)$
A	A	0	1
B	A	0	1
D	A	0	0
A	B	0	1
B	B	1	1
D	B	0	0
A	D	0	1
B	D	1	1
D	D	0	0

Definition:

For the given intuitionistic fuzzy predicates $P_i(x)$ and $P_j(y)$, the lower and upper disjunctions $\underset{\alpha}{\vee}$ and $\overset{\alpha}{\vee}$ are defined as

$P_i(x) \underset{\alpha}{\vee} P_j(y) = P_{i,\alpha}(x) \vee P_{j,\alpha}(y)$ and

$P_i(x) \overset{\alpha}{\vee} P_j(y) = P_i^{\alpha}(x) \vee P_j^{\alpha}(y)$ respectively.

Here, $(\underset{\alpha}{\vee}, \overset{\alpha}{\vee})$ is called **rough disjunction**.

Example:

Let the universe of discourse having U={a,b,c,d,e,f} with the partition Ψ={{a,c,e},{b,f},{d}}. Consider the intuitionistic fuzzy predicates P and Q defined on U which are given by

P=((0.2,0.79),(0.6,0.39),(0.5,0.49),(0.4,0.59),(0.3,0.68),(0.7,0.28)) and Q=((0.4,0.58),(0.6,0.38), (0.3,0.69),(0.6,0.39),(0.5,0.47),(0.8,0.19)).

Let α=0.45. Then P[α]={b,c,f} and Q[α]={b,d,e,f}. Hence, P_α={b,f}, P^α={a,b,c,e,f}, Q_α={b,d,f} and $Q^\alpha=\Psi$.

x_1	x_2	$P(x_1)\underset{\alpha}{\vee}Q(x_2)$	$P(x_1)\overset{\alpha}{\vee}Q(x_2)$
A	A	0	1
B	A	1	1
D	A	0	1
A	B	1	1
B	B	1	1
D	B	1	1
A	D	1	1
B	D	1	1
D	D	1	1

Definition:

For the given intuitionistic fuzzy predicate $P_i(x)$, the lower and upper m-negations $\tau_{m(\alpha)}$ and $\tau^{m(\alpha)}$ are defined as

$\tau_{m(\alpha)}P_i(x)=(m-negP_i(x))_\alpha$ and

$\tau^{m(\alpha)}P_i(x)=(m-negP_i(x))^\alpha$ respectively.

Here, $(\tau_{m(\alpha)},\tau^{m(\alpha)})$ is called **rough m-negation**.

Example:

Let the universe of discourse having U={a,b,c,d,e,f} with the partition Ψ={{a,c,e},{b,f},{d}}. Consider the intuitionistic fuzzy predicates P and Q defined on U which are given by

P=((0.2,0.79),(0.6,0.39),(0.5,0.49),(0.4,0.59),(0.3,0.68),(0.7,0.28)). Let α=0.45. Then P[α]={b,c,f} gives P_α={b,f} and P^α={a,b,c,e,f}.

x_1	$\tau_{m(\alpha)}P(x_1)$	$\tau^{m(\alpha)}P(x_1)$
a	1	1
b	0	0
d	1	1

Definition:

For the given intuitionistic fuzzy predicates $P_i(x)$ and $P_j(y)$, the lower and upper m- implications $\xrightarrow[m(\alpha)]{}$ and $\xrightarrow{m(\alpha)}$ are defined as

$P_i(x) \xrightarrow[m(\alpha)]{} P_j(y) = \tau_{m(\alpha)} P_i\ (x) \vee P_{j,\alpha}(y)$ and

$P_i(x) \xrightarrow{m(\alpha)} P_j(y) = \tau^{m(\alpha)} P_i\ (x) \vee P_j^{\alpha}\ (y)$ respectively.

Here, ($\xrightarrow[m(\alpha)]{}$, $\xrightarrow{m(\alpha)}$) is called **rough m-implication**.

Example:

Let the universe of discourse having U={a,b,c,d,e,f} with the partition Ψ={{a,c,e},{b,f},{d}}. Consider the intuitionistic fuzzy predicates P and Q defined on U which are given by

P=((0.2,0.79),(0.6,0.39),(0.5,0.49),(0.4,0.59),(0.3,0.68),(0.7,0.28)) and Q=((0.4,0.58),(0.6,,0.38), (0.3,0.69),(0.6,0.39),(0.5,0.47),(0.8,0.19)).

Let α=0.45. Then P[α]={b,c,f} and Q[α]={b,d,e,f}. Hence, P_α={b,f}, P^α={a,b,c,e,f}, Q_α={b,d,f} and Q^α=Ψ.

x_1	x_2	$P(x_1)\underset{\alpha}{\rightarrow}Q(x_2)$	$P(x_1)\overset{\alpha}{\rightarrow}Q(x_2)$
A	A	1	1
B	A	0	1
D	A	1	1

A	B	1	1
B	B	1	1
D	B	1	1
A	D	1	1
B	D	1	1
D	D	1	1

Definition:

For the given intuitionistic fuzzy predicates $P_i(x)$ and $P_j(y)$, the lower and upper m- bi-implications $\xleftrightarrow[m(\alpha)]{}$ and $\xleftrightarrow{m(\alpha)}$ are defined as

$P_i(x) \xleftrightarrow[m(\alpha)]{} P_j(y) =$

$[P_i(x) \xrightarrow[m(\alpha)]{} P_j(y)] \wedge [P_j(y) \xrightarrow[m(\alpha)]{} P_i(x)]$ and $P_i(x) \xleftrightarrow{m(\alpha)} P_j(y) =$

$[P_i(x) \xrightarrow{m(\alpha)} P_j(y)] \wedge [P_j(y) \xrightarrow{m(\alpha)} P_i(x)]$ respectively.

Here, $(\xleftrightarrow[m(\alpha)]{}, \xleftrightarrow{m(\alpha)})$ is called **rough m- bi-implication**.

Example:

Let the universe of discourse having U={a,b,c,d,e,f} with the partition Ψ={{a,c,e},{b,f},{d}}. Consider the intuitionistic fuzzy predicates P and Q defined on U which are given by
P=((0.2,0.79),(0.6,0.39),(0.5,0.49),(0.4,0.59),(0.3,0.68),(0.7,0.28)) and
Q=((0.4,0.58),(0.6,,0.38), (0.3,0.69),(0.6,0.39),(0.5,0.47),(0.8,0.19)).
Let α=0.45. Then $P[\alpha]$={b,c,f} and $Q[\alpha]$={b,d,e,f}. Hence, P_α={b,f}, P^α={a,b,c,e,f}, Q_α={b,d,f} and Q^α=Ψ.

x_1	x_2	$P(x_1)\underset{\alpha}{\leftrightarrow}Q(x_2)$	$P(x_1)\overset{\alpha}{\leftrightarrow}Q(x_2)$
A	A	1	1

B	A	0	1
D	A	1	1
A	B	0	1
B	B	1	1
D	B	0	0
A	D	0	1
B	D	1	1
D	D	0	0

Definition:

For the given intuitionistic fuzzy predicate $P_i(x)$, the lower and upper nm-negations $\tau_{nm(\alpha)}$ and $\tau^{nm(\alpha)}$ are defined as

$\tau_{nm(\alpha)} P_i(x) = (nm - negP_i(x))_\alpha$ and

$\tau^{nm(\alpha)} P_i(x) = (nm - negP_i(x))^\alpha$ respectively.

Here, $(\tau_{nm(\alpha)}, \tau^{nm(\alpha)})$ is called **rough nm-negation**.

Now using rough nm-negation, we can define rough nm-implication and rough nm-bi-implication.

CHAPTER 10

NAÏVE BAYESIAN ROUGH CLASSIFICATION BY FUZZY AND INTUITIONISTIC FUZZINESS

Slezak initiated a study on Bayesian Rough Set model [30, 31] and Yiyu Yao et. Al [35] discussed the Naïve Bayesian Rough Set Model. In this chapter, we extend the rough indexing to the information systems by incorporating the Probabilistic Naïve Bayesian Rough Set Model.

10.1. Decision-Theoretic and Probabilistic Rough Sets

Rough Sets theory defines two-way approximations for a given input that is lower and upper approximations. For a given finite universe of discourse having U and an equivalence relation E, the equivalence class of any x∈U is described to be $[x]=\{y\in U / xEy\}$. The family of equivalence classes $U/E=\{[x]_E \mid x\in U\}$ is a partition of the universe U. For the provided concept C, Pawlak defined the lower approximation as $\underline{apr}_E(C)=\{x\in U/[x]_E\subseteq C\}$ and the upper approximation as $\overline{apr}_E(C)=\{x\in U/[x]_E\cap C\neq\Phi\}$.

For the provided concept C, three disjoint regions, namely positive, negative and boundary regions are defined below:

Positive Region: $POS_E(C)=\{x\in U/[x]_E\subseteq C\}$ **(10.1)**

Boundary : $BND_E(C)=\{x\in U/[x]_E\cap C\neq\Phi\wedge[x]_E\not\subseteq C\}$ **(10.2)**

Negative region: $NEG_E(C) = \{x \in U / [x]_E \cap \in \quad \Phi\}$ **(10.3)**

As Pawlak's model is restrictive, many researchers concentrated on generalizing this approach towards generalized rough set model, parameterized rough set model and probabilistic rough set model.

In 1994, Pawlak and Skowron stated rough membership function by considering degrees of overlap between equivalence classes and a concept C to be approximated and it is observed as the conditional probability of an object which belongs to C provided that the object is in [x] (for simplicity, we denote $[x]_E$ with [x]) which is given as $\Pr\left(C/[x]\right) = \frac{|C \cap [x]|}{|[x]|}$

The positive, boundary and negative regions are stated as follows by using the definition above:

$$POS(C) = \left\{x \in U \,/\, \Pr\left(C/[x]\right) = 1\right\} \quad \textbf{(10.4)}$$

$$BND(C) = \left\{x \in U \,/\, 0 < \Pr\left(C/[x]\right) < 1\right\} \quad \textbf{(10.5)}$$

$$NEG(C) = \left\{x \in U \,/\, \Pr\left(C/[x]\right) = 0\right\} \quad \textbf{(10.6)}$$

In 2009, Greco et.al exchange views on the parameterized rough set model [23] by generalizing the above-said definitions. In this model to define the probabilistic regions, two thresholds namely α and β are used and the positive, boundary and negative regions are modified as follows:

$$POS_{(\alpha,\beta)}(C) = \left\{x \in U \,/\, \Pr\left(C/[x]\right) \geq \alpha\right\} \quad \textbf{(10.7)}$$

$$BND_{(\alpha,\beta)}(C) = \left\{x \in U \,/\, \beta < \Pr\left(C/[x]\right) < \alpha\right\} \quad \textbf{(10.8)}$$

$$NEG_{(\alpha,\beta)}(C) = \left\{x \in U \,/\, \Pr\left(C/[x]\right) \leq \beta\right\} \quad \textbf{(10.9)}$$

For any object x in U, these Probabilistic regions will lead through a three-way decisions that is acceptance, deferment and rejection respectively. However, it is easy to compute the probability of the existence of a category [x] for a provided concept C using $\Pr\left([x]/C\right) = \frac{|[x] \cap C|}{|C|}$ in many cases.

The Positive, Boundary and Negative Regions by using Baye's Theorem are given by

$$POS^{B}_{(\alpha',\beta')}(C) = \left\{ x \in U \,/\, \log \frac{\Pr([x]/C)}{\Pr([x]/C^{c})} \geq \alpha' \right\} \quad \textbf{(10.10)}$$

$$BND^{B}_{(\alpha',\beta')}(C) = \left\{ x \in U \,/\, \beta' < \log \frac{\Pr([x]/C)}{\Pr([x]/C^{c})} < \alpha' \right\} \quad \textbf{(10.11)}$$

$$NEG^{B}_{(\alpha',\beta')}(C) = \left\{ x \in U \,/\, \log \frac{\Pr([x]/C)}{\Pr([x]/C^{c})} \leq \beta' \right\} \quad \textbf{(10.12)}$$

$$where \quad \alpha' = \log \frac{\Pr(C^{c})}{\Pr(C)} + \log \frac{\alpha}{1-\alpha}$$

$$and \quad \beta' = \log \frac{\Pr(C^{c})}{\Pr(C)} + \log \frac{\beta}{1-\beta}$$

10.2. Naïve Bayesian Probabilistic Rough Sets Model for a Fuzzy Concept

In the above two segments, the threshold α has been used with the same value, for non similar purposes, to make homogeneity, here, we supplant the threshold α to obtain a Strong Cut on fuzzy sets with δ.

Thus for a given fuzzy concept F with the threshold δ, the probabilistic positive, boundary and negative regions are correspondingly defined on the approximation space U/E as

$$POS_{\delta}(F) = \left\{ x \in U \,/\, \Pr\left(F[\delta] / [x] \right) = 1 \right\} \quad \textbf{(10.13)}$$

$$BND_{\delta}(F)=\left\{x\in U\,/\,0<\Pr\left(F[\delta]\big/[x]\right)<1\right\}$$ **(10.14)**

$$NEG_{\delta}(F)=\left\{x\in U\,/\,\Pr\left(F[\delta]\big/[x]\right)=0\right\}$$ **(10.15)**

The regions of the parameterized rough sets model for the parameters α and β are given by

$$POS_{(\alpha,\beta,\delta)}(F)=\left\{x\in U\,/\,\Pr\left(F[\delta]\big/[x]\right)\geq\alpha\right\}$$ **(10.16)**

$$BND_{(\alpha,\beta,\delta)}(F)=\left\{x\in U\,/\,\beta<\Pr\left(F[\delta]\big/[x]\right)<\alpha\right\}$$ **(10.17)**

$$NEG_{(\alpha,\beta,\delta)}(F)=\left\{x\in U\,/\,\Pr\left(F[\delta]\big/[x]\right)\leq\beta\right\}$$ **(10.18)**

α and β,

and the Naïve Bayesian Rough Sets Model regions are given by

$$POS^{B}_{(\alpha',\beta',\delta)}(F)=\left\{x\in U/\log\frac{\Pr\left([x]/F[\delta]\right)}{\Pr\left([x]/(F[\delta])^{C}\right)}\geq\alpha'\right\}$$ **(10.19)**

$$BND^{B}_{(\alpha',\beta',\delta)}(F)=\left\{x\in U/\beta'<\log\frac{\Pr\left([x]/F[\delta]\right)}{\Pr\left([x]/(F[\delta])^{C}\right)}<\alpha'\right\}$$ **(10.20)**

$$NEG^{B}_{(\alpha',\beta',\delta)}(F)=\left\{x\in U\,/\,\log\frac{\Pr\left([x]/F[\delta]\right)}{\Pr\left([x]/(F[\delta])^{C}\right)}\leq\beta'\right\}\quad where\quad \alpha'=\log\frac{\Pr(C^{c})}{\Pr(C)}+\log\frac{\alpha}{1-\alpha}$$

$$and\quad \beta'=\log\frac{\Pr(C^{c})}{\Pr(C)}+\log\frac{\beta}{1-\beta}$$ **(10.21)**

10.3. Rough Indices

Consider U be the universe of discourse and α will be any value in (0,1). Consider X={W_1,W_2,…,W_n} will be any partition illustrate on U. For any fuzzy set A define $A[\alpha]=\{x\in U/\ \mu_A(x)>\alpha\}$ where α is taken from R-domain satisfying the property that $dil^n(\alpha)$ and $con^n(\alpha)$ are the associates of R-Domain for any positive integer n [dil for dilation and con for concentration]. The lower and upper approximations A_α and A^α are given by $A_\alpha=(A[\alpha])_\sim$ and $A^\alpha=(A[\alpha])^\sim$ correspondingly.

Consider M signifies the largest number under consideration such that n+M is constantly positive and n-M is constantly negative for any integer n.

Now, we shall modify this algorithm for a three-way approach on rough sets are defined as follow:

Algorithm Three_Way_ rough index (x,A,δ)

//Algorithm returns the Three_Way_rough index of x

1. Let x_index be an integer initialized to 0
2. Pick the equivalence class K containing x.

        If $\mu_A(y)=0$ for all $y\in K$
        begin
                x_index=-M
                goto 6
        end
    end

3. If $\mu_A(x)=1$

    begin
        If $\mu_A(y)=1$ for all $y\in K$
        begin
                x_index=M
                goto 6
        end
    end

```
4. compute POSδ(A), BNDδ(A) and NEGδ(A)
5. If x∈ POSδ(A)
   begin
   x_index=M
   while (x∈ POSδ(A))
     begin
          α=dil(δ)          //dilation of δ
          x_index=x_index+1
          compute POSδ(A)
     end
   end
   else
          if x∈ NEGδ(A)
          begin
          x_index=-M
          while (x∈ NEGδ(A))
            begin
               δ=con(δ)//concentration of δ
               x_index=x_index-1
               compute NEGδ(A)
             end
          end
          else
          Let γ=δ
          compute NEGγ(A)
          while (x∉(POSδ(A)∪NEGγ(A)))
            begin
              δ=con(δ) // concentration of δ
              γ=dil(γ)      // dilation of γ
```

compute $POS_{\delta}(A) \cup NEG_{\gamma}(A)$

x_index=x_index+1

end

if $x \in POS_{\delta}(A)$ then

x_index= - x_index

end

6. return x_index

Now, we parameterize the algorithm using parameters α and β.

Algorithm Naïve Bayesian_ rough index $(x,A,\alpha,\beta,\delta)$

//Algorithm returns Naïve Bayesian_rough index of x

1. Let x_index be an integer initialized to 0
2. Pick the equivalence class K containing x.

If $\mu_A(y)=0$ for all $y \in K$

begin

x_index=-M

goto 6

end

end

3. If $\mu_A(x)=1$

begin

If $\mu_A(y)=1$ for all $y \in K$

begin

x_index=M

goto 6

end

end

4. compute $POS^B_{(\alpha',\beta',\delta)}(A)$, $BND^B_{(\alpha',\beta',\delta)}(A)$ and $NEG^B_{(\alpha',\beta',\delta)}(A)$

5. If $x \in POS^B_{(\alpha',\beta',\delta)}(A)$

   begin

   x_index=M

   while ($x \in POS^B_{(\alpha',\beta',\delta)}$ (A))

   begin

   α=dil(δ) //dilation of δ

   x_index=x_index+1

   compute $POS^B_{(\alpha',\beta',\delta)}(A)$

   end

   end

else

if $x \in NEG^B_{(\alpha',\beta',\delta)}(A)$

begin

x_index=-M

while ($x \in NEG^B_{(\alpha',\beta',\delta)}(A)$)

begin

δ=con(δ)//concentration of δ

x_index=x_index-1

compute $NEG^B_{(\alpha',\beta',\delta)}(A)$

end

end

else

Let $\gamma=\delta$

compute $NEG_\gamma(A)$

while

($x \notin (POS^B_{(\alpha',\beta',\delta)}(A) \cup NEG^B_{(\alpha',\beta',\gamma)}(A))$)

begin

δ=con(δ) // concentration of δ

γ=dil(γ) // dilation of γ

compute

$POS^{B}_{(\alpha',\beta',\delta)}(A) \cup NEG^{B}_{(\alpha',\beta',\gamma)}(A)$

x_index=x_index+1

end

if $x \in POS^{B}_{(\alpha',\beta',\delta)}$ (A) then

x_index= - x_index

end

6. return x_index

10.4. Naïve Bayesian Indexing in Information System with Fuzzy Decision Attribute

According to the perspective of Z.Pawlak, any information system is given by T=(U, A, C, D), where U is the universe of discourse, A is a set of primitive attributes, C and D are the subsets of A called condition and decision features respectively [in few of the information systems, C and D may not exist].

Consider an information system with conditional attributes $C=\{a_1,a_2,\ldots,a_n\}$ and decision attributes $\{d_1,d_2,\ldots,d_s\}$ with the records $U=\{x_1,x_2,\ldots,x_m\}$. The indiscernibility relation is given by $x_i \approx_{a_k} x_j$ (read as x_i is related to x_j with respect to a_k) for any index key 'a' in C, if and only if $a_k(x_i)=a_k(x_j)$. It is clearly noted that this indiscernibility relation partitions the universe of discourse U. However, the selection process of the appropriate minimal attributes [reducts] for effectiveness is not discussed in this chapter.

For example, consider the decision table with C={a,b,c,d} and D={E}.

Table 10.1 Decision Table

	A	B	c	D	E
x_1	1	0	2	1	1
x_2	1	0	2	0	1
x_3	1	2	0	0	2
x_4	1	2	2	1	0
x_5	2	1	0	0	2
x_6	2	1	1	0	2
x_7	2	1	2	1	1

Consider ‘c’ as the index key. As x_1,x_2,x_4,x_7 have the values 2; x_3,x_5 have the values 0 and x_6 has the value 1.So the partition on U with reference to c can be stated as $\{\{x_1,x_2,x_4,x_7\},\{x_3,x_5\},\{x_6\}\}$.

However, we can find many information systems with fuzzy decision attributes in real-time systems and thus the scope of the algorithms discussed above would be applicable to those information systems. In this place, the Naïve Bayesian rough indexing of the data can be derivative from the fuzzy decision attribute as explained in the past session.

For instance, concede knowledge representation of the information system with C={a,b,c,d} and D={E} where E is of fuzzy natured.

Table 10.2 Knowledge Representation

	A	B	c	D	$\mu_E(x_i)$
x_1	1	0	2	1	0.45
x_2	1	0	2	0	0.7
x_3	1	2	0	0	0.65
x_4	1	2	2	1	0.1
x_5	2	1	0	0	0.91
x_6	2	1	1	0	0.6
x_7	2	1	2	1	0.35

Let 'c' as the index key, the partition acquired is $\{\{x_1,x_2,x_4,x_7\}, \{x_3,x_5\},\{x_6\}\}$. Let δ=0.5. Here, $E[\delta] = \{x_2,x_3,x_5,x_6\}$. For the provided α and β, the Naïve Bayesian indexing algorithm would be carried out further.

10.5. Naïve Bayesian Probabilistic Rough Sets Model for an Intuitionistic Fuzzy Concept

The used thresholds α and β are same in the both of the above sections, hereafter, on intuitionistic fuzzy sets we use (δ,γ) cuts.

Thus, the probabilistic positive, boundary and negative regions for a given intuitionistic fuzzy concept F with the thresholds δ and γ are correspondingly defined on the approximation space U/E as

$$POS_{(\delta,\gamma)}(F)=\left\{x\in U / \Pr\left(F[\delta,\gamma] / [x]\right)=1\right\} \quad \textbf{(10.22)}$$

$$BND_{(\delta,\gamma)}(F)=\left\{x\in U / 0<\Pr\left(F[\delta,\gamma] / [x]\right)<1\right\} \quad \textbf{(10.23)}$$

$$NEG_{(\delta,\gamma)}(F)=\left\{x\in U / \Pr\left(F[\delta,\gamma] / [x]\right)=0\right\} \quad \textbf{(10.24)}$$

The regions of the parameterized rough sets model for the given parameters α and β, are given by

$$POS_{(\alpha,\beta,\delta,\gamma)}(F)=\left\{x\in U / \Pr\left(F[\delta,\gamma] / [x]\right)\geq\alpha\right\} \quad \textbf{(10.25)}$$

$$BND_{(\alpha,\beta,\delta,\gamma)}(F)=\left\{x\in U / \beta<\Pr\left(F[\delta,\gamma] / [x]\right)<\alpha\right\} \quad \textbf{(10.26)}$$

$$NEG_{(\alpha,\beta,\delta,\gamma)}(F)=\left\{x\in U / \Pr\left(F[\delta,\gamma] / [x]\right)\leq\beta\right\} \quad \textbf{(10.27)}$$

and the Regions of Naïve Bayesian Rough Sets Model are provided by

$$POS^{B}_{(\alpha',\beta',\delta,\gamma)}(F)=\left\{x\in U / \log\frac{\Pr\left([x]/F[\delta,\gamma]\right)}{\Pr\left([x]/(F[\delta,\gamma])^{C}\right)}\geq\alpha'\right\} \quad \textbf{(10.28)}$$

$$BND^{B}_{(\alpha',\beta',\delta,\gamma)}(F)=\left\{x\in U / \beta'<\log\frac{\Pr\left([x]/F[\delta,\gamma]\right)}{\Pr\left([x]/(F[\delta,\gamma])^{C}\right)}<\alpha'\right\} \quad \textbf{(10.29)}$$

$$NEG^{B}_{(\alpha',\beta',\delta,\gamma)}(F)=\left\{x\in U / \log\frac{\Pr\left([x]/F[\delta,\gamma]\right)}{\Pr\left([x]/(F[\delta,\gamma])^{C}\right)}\leq\beta'\right\} \quad where \quad \alpha'=\log\frac{\Pr(C^{c})}{\Pr(C)}+\log\frac{\alpha}{1-\alpha}$$

$$and \quad \beta'=\log\frac{\Pr(C^{c})}{\Pr(C)}+\log\frac{\beta}{1-\beta} \quad \textbf{(10.30)}$$

10.6. Rough Indices

Algorithm Naïve Bayesian_ rough index ($x,A,\alpha,\beta,\delta,\gamma$)

//Algorithm returns the Naïve Bayesian_rough index of x

1. Consider an integer x_index initialized to 0
2. Choose the equivalence class K containing x.
3. If $\mu_A(y)=0$ and $\gamma_A(y)=1$ for all $y \in K$

   x_index=-M

   goto 7
4. If $\mu_A(x)=1$ and $\gamma_A(y)=0$ for all $y \in K$

   x_index=M

   goto 7
5. compute $POS^B_{(\alpha',\beta',\ \delta,\gamma)}(A)$, $BND^B_{(\alpha',\beta',\delta,\gamma)}(A)$ and $NEG^B_{(\alpha',\beta',\delta,\gamma)}(A)$
6. If $x \in POS^B_{(\alpha',\beta',\delta,\gamma)}(A)$

```
    begin
        x_index=M
        while (x∈ POS^B_(α',β', δ,γ) (A))
                begin
                    δ=dil(δ)        //dilation of δ
                    γ=dil(γ)        //dilation of γ
                    x_index=x_index+1
                    compute POS^B_(α',β', δ,γ)(A)
                end
        end
      end
  else if x∈ NEG^B_(α',β', δ,γ)(A)
    begin
        x_index=-M
        while (x∈ NEG^B_(α',β', δ,γ)(A))
```

begin

δ=con(δ) //concentration of δ

γ=con(γ) //Concentration of γ

x_index=x_index-1

compute $NEG^{B}_{(\alpha',\beta', \delta,\gamma)}(A)$

end

end

end

else

Let $\pi_1=\delta$; $\pi_2=\delta$; $\theta_1=\gamma$;$\theta_2=\gamma$

while ($x \notin (POS^{B}_{(\alpha',\beta',\pi1,\theta1)}(A) \cup NEG^{B}_{(\alpha',\beta', \pi2,\theta2)}(A))$)

begin

π_1=Con(π_1);

θ_1=Con(θ_1);

π_2=dil(π_2);

θ_2=dil(θ_2);

x_index=x_index+1

end

if $x \in POS^{B}_{(\alpha',\beta', \pi1,\theta1)}(A)$

x_index= - x_index

end

7. return x_index

10.7. Naïve Bayesian Indexing in Information System with Intuitionistic Fuzzy Decision Attributes

Consider T=(U, A, C, D) as an information system, where U is the universe of discourse, A is a set of primitive attributes, C and D are the subsets of A which are termed as condition and decision features respectively [in few of the information systems, C and D may not exist].

Let $C=\{a_1,a_2,\ldots,a_n\}$ and $D=\{d_1,d_2,\ldots,d_s\}$ with the records $U=\{x_1,x_2,\ldots,x_m\}$. The indiscernibility relation is given by $x_i \approx_{a_k} x_j$ (read as x_i is related to x_j with respect to a_k) for any index key 'a' in C, if and only if $a_k(x_i)=a_k(x_j)$. It is clearly noted that this indiscernibility relation partitions the universe of discourse U. The selection process of the appropriate minimal attributes [reducts] for effectiveness is not discussed in this chapter.

Consider the decision table with C={a,b,c,d} and D={E}.

Table 10.3 Decision Table

	A	B	C	d	E
x_1	1	0	2	1	1
x_2	1	0	2	0	1
x_3	1	2	0	0	2
x_4	1	2	2	1	0
x_5	2	1	0	0	2
x_6	2	1	1	0	2
x_7	2	1	2	1	1

Consider the index key as ‘c’. As x_1,x_2,x_4,x_7 have the values 2; x_3,x_5 have the values 0 and x_6 has the value 1. Thus, the partition on U with respect to c can be defined as $\{\{x_1,x_2,x_4,x_7\},\{x_3,x_5\},\{x_6\}\}$.

We can discover several information systems with fuzzy or intuitionistic fuzzy decision attributes in our real-time systems. As discussed earlier, with intuitionistic fuzzy decision attributes, for any information system, the Naïve Bayesian rough indices algorithm can be applied.

10.8. Experimental Analysis

Let a data table with C={a,b,c,d} and D={E} where E is intuitionistic fuzzy natured.

Table 10.4 Data Table for experimental analysis

	A	b	c	d	$\mu_E(x_i)$	$\eta_E(x_i)$
x_1	1	0	2	1	0.45	0.54
x_2	1	0	2	0	0.7	0.2
x_3	1	2	0	0	0.65	0.3
x_4	1	2	2	1	0.1	0.6
x_5	2	1	0	0	0.91	0.03
x_6	2	1	1	0	0.6	0.31

The partition obtained is $\{\{x_1,x_2,x_4,x_7\},\{x_3,x_5\},\{x_6\}\}$, on considering ‘c’ as the index key. Let $\delta=0.5$ and $\gamma=0.39$. Here, $E[\delta,\gamma]=\{x_2,x_3,x_5,x_6\}$.

Let $\alpha'=0.999$ and $\beta'=0$. Choose the element x_1.

Here, $x_1 \in POS(E)$ where POS represents the Positive region as quoted in the algorithm and hence, initially x_index is assigned M. On applying the dilation on δ and γ, we obtain $\delta=0.7071$ and $\gamma=0.1541$, and hence $E[\delta,\gamma]=\{x_5\}$. As $x_1 \notin POS(E)$, the algorithm returns the index of x_1 to be M+1.

CHAPTER 11

CONCLUSION

The work deals with classifying the records in an information system in which there is fuzzy or intuitionistic decision attributes and provides various indexing algorithms. An algorithm for classifying has been designed by using the hedges of the fuzzy / intuitionistic fuzzy attribute, the approximations using rough sets and threshold variables. The rough indexing algorithm for the three-way approach is modified and extended the rough indexing to the information systems by incorporating the Probabilistic Naïve Bayesian Rough Set Model. This work has been organized into Ten chapters. In Soft Computing, various types of ambiguity prevail. Making decisions under these uncertainties is risky. The algorithms and theorems developed in this work help in developing tools with precision. In recent days, there are several research works carried out on rough sets under fuzzy environment and the present study is an attempt to give an algebraic treatment of rough and fuzzy approaches. This work opens a way for engineers and technicians to develop Soft Computing tools for their applications.

REFERENCES

1. Abraham, S. Henry, F.Korth and Sudarshan, S. "Database System Concepts", McGrawHill Companies, Inc.1997.

2. Alicja, M. and Leszek, R. "Fuzziness in information systems", Electronic notes in theoretical Computer Science 82, No: 4, 2003; url:http://www.elsevier.nl/locate/entcs/volume82.html.

3. Atanassov, KT. "Intuitionistic Fuzzy Sets, Fuzzy Sets and Systems 20", pp:87-96, 1986.

4. Balasubramaniam, J. and Jagan Mohan Rao, C. "On the distributivity of Implication operators over T and S norms", IEEE Transactions on Fuzzy Systems, Vol 12, No 2, pp 194-198, April 2004.

5. Biswas, R. "On Rough Sets and Fuzzy Rough Sets", Bulletin of the Polish Academy of Sciences, Mathematics; Vol42 ; No: 4, pp 343-349, 1994.

6. Biswas, R. "On Rough Fuzzy Sets", Bulletin of the Polish Academy of Sciences, Mathematics; Vol 42; No: 4, pp 352-355, 1994.

7. Buszkowski, W. and Orlowska E. "On the logic of database dependencies", Bull. Polish Sci. Math., Vol 34, pp 345-354, 1986.

8. Baldwin, J.F. "Fuzzy logic and fuzzy reasoning, Fuzzy Reasoning and Its Applications", E.H. Mamdani and B.R. Gaines (eds.), London: Academic Press, 1981.

9. David, M. Kroenke. "Database Processing: Fundamentals, Design, Implementation"; Galgotia Publications Pvt. Ltd., 2nd edition, 1994.

10. De, S.K., Biswas, R. and Roy, A.R. "Rough Sets on Fuzzy Approximation Space", Journal of Fuzzy Mathematics, 11, pp 1-15, 2003.

11. Dubois, D. and Prade, H. "Rough Fuzzy sets and Fuzzy Rough Sets", International Journal of General Systems, 17, pp 191-209, 1989.

12. Dubois, D. and Prade, H. “Putting Rough sets and Fuzzy sets together, Intelligent Decision Support”, Hand book of Applications and Advances of the Rough sets, Kluwer Academic Publications, Dordrecht, 1992.

13. Ganesan, G. and Raghavendra Rao, C. “Feature Selection using Fuzzy Decision Attributes”, Journal of INFORMATION, Vol 9, No3, pp:381-394, 2006.

14. Ganesan, G. and Raghavendra Rao, C. “Rough Set: Analysis of Fuzzy Sets using thresholds”, Computational Mathematics, Narosa, India,. pp: 81-87, 2005.

15. Ganesan, G. and Raghavendra Rao, C. “Reducts in Information System with Conditional Decision Attributes”, Computing and Mathematical Modelling, Narosa, India, pp:40-46, 2006.

16. Ganesan, G., Raghavendra Rao, C. and Latha, D. “Rough index induced by a fuzzy set using two thresholds”, proceedings of the International Conference on Soft Computing, BIHER, Chennai ,India, pp: 164-169, 2004.

17. Ganesan, G. and Raghavendra Rao, C. “Relation between Rough Fuzzy Sets and Rough Sets on Fuzzy Sets using thresholds”, proceedings of the National Conference on the emerging trends in Pure and Applied Mathematics, Palayamkottai, India, pp: 102-108, 2005.

18. Ganesan, G., Raghavendra Rao, C. and Latha, D. “An overview of rough sets”, proceedings of the National Conference on the emerging trends in Pure and Applied Mathematics, Palayamkottai, India, pp: 70-76, 2005.

19. Ganesan, G., Latha, D., and Raghavendra Rao, C. “Rough Index in Information System with Fuzziness in Decision Attributes”, International journal of Fuzzy Mathematics, Vol: 16, No: 4, 2008.

20. Ganesan, G. “Analysis of Intuitionistic Fuzzy Sets using Thresholds-An Approach of Rough Sets”, International Journal of Computing and Mathematical Applications, Vol: 2, pp: 263-270, 2008.

21. Ganesan, G. “Intuitionistic Rough Fuzziness and Generalizations”, Journal of Convergence in Information Technology, Vol:3, No:3, pp:82-88, 2008.

22. George, J.Klir. and Bo, Yuan, “Fuzzy Sets and Fuzzy Logic Theory and Applications”, Prentice-Hall of India Pvt Ltd.,1997.

23. Greco, S., Matarazzo, B. and Slowinski, R. “Parameterized rough set model using rough membership and Bayesian confirmation measures”, International Journal of Approximate Reasoning, 49, 285-300, 2009.

24. Joe, L.Mott, Abraham, Kandel and Theodore, P. Baker, “Discrete Mathematics for Computer Scientists and Mathematicians”, Prentice Hall of India Pvt Ltd., 1999.

25. Kohavi, R. “Useful feature subsets and Rough set reducts”, Proceedings, Third International Workshop on Rough Set and Soft Computing, pp 310-317, 1994.

26. Nakamura, A. “Fuzzy Rough Sets, Note on Multiple valued Logic”, Japan 9(8), pp 1-8, 1988.

27. Ning, Zhong, Juzhen, Dong and Setsuo, Ohsuga, “Using Rough Sets with Heuristics for Feature Selection”, Journal of Intelligent Information systems, Vol 16, pp.199-214, 2001.

28. Ross, T.J. “Fuzzy Logic with Engineering Applications”, McGraw Hill International Editions, 1997.

29. Skowron, A. and Rauszer, C. “The discernibility matrices and functions in information systems”, Fundamenta Informaticae 15(2), pp.331-362, 1991.

30. Slezak, D. “Rough Sets and Baye’s Factor”, LNCS Transactions on Rough Sets III, Springer Verlag, Heidelberg, 3400, 2005.

31. Slezak, D. and Ziarko, W. “The investigation of the Bayesian rough set model”, International Journal of Approximate Reasoning, 40, 81-91, 2005.

32. Starzyk, J., Nelson, D.E. and Sturtz, K. “Reduct Generation in Information Systems”, Bulletin of International Rough Set Society, 3 (1/2), 1999.

33. Wong, S.K.M. and Ziarko, W. “Comparison of the probabilistic approximate classification and the fuzzy set model, Fuzzy Sets and Systems”, 21, pp 357-362, 1987.

34. Yao, Y.Y. “Generalized Rough Set models, Rough Sets in Knowledge discovery”, Physica-Verlag, Heidelberg, pp 286-318, 1998.

35. Yiyu, Yao and Bing, Zhou, "Naive Bayesian Rough Sets", Proceedings of RSKT 2010, LNAI 6401, pp. 719-726, 2010.

36. Zadeh, L. "Fuzzy Sets", Journal of Information and Control, 8, pp. 338-353, 1965.

37. Zadeh, L. "A fuzzy set theoretic interpretation of linguistic hedges", J. Cybern., Vol. 2, no. 2, pp. 4-34, 1972.

38. Zdzislaw, Pawlak, "Information systems - theoretical foundations, Information Systems", Vol. 6, pp.205-218, 1981.

39. Zdzislaw, Pawlak, "Rough sets, International Journal of Computer and Information Sciences", 11, 341-356, 1982.

40. Zdzislaw, Pawlak, "On rough dependency of attributes in information systems", Bull. Polish Acad. Sci. Tech. Vol. 33, pp.481-485, 1985.

41. Zdzislaw, Pawlak, "Rough Sets-Theoretical Aspects and Reasoning about Data", Kluwer Academic Publications, 1991.

42. Ziarko, Z. "Variable Precision Rough Set Model", Jr. of Computer and System Sciences, pp:39-59,1993.

PUBLICATIONS

1. Rekha, D. and Thangadurai, K. “Probabilistic Rough Indices in Information Systems under Intuitionistic Fuzziness”, International Journal on Recent and Innovation Trends in Computing and Communication, Vol: 2, Issue: 8, pp.2117-2121, 2014.

2. Latha, D., Rekha, D., Thangadurai, K. and Ganesan, G. “Probabilistic Rough Classification in Information Systems under Intuitionistic Fuzziness”, IFRSA International Journal of Data Warehousing & Mining, Vol: 3, Issue: 2, pp. 82-86, 2013.

3. Latha, D., Rekha, D., Thangadurai, K. and Ganesan, G. “Probabilistic Rough Classification in Information Systems with Fuzzy Decision Attributes”, International Journal on Recent and Innovation Trends in Computing and Communication, Vol: 1, Issue: 6, pp.547-552, 2013.