Université de Tunis
Institut Supérieur de Gestion
École Doctorale Sciences de Gestion

Belief Rough Set Classifier

THESE
En vue de l'obtention du Doctorat
En Informatique de Gestion

Présentée et soutenue publiquement par:

Salsabil TRABELSI

Le 19 septembre 2011

Membres du Jury:

♦ ***M. Khaled MELLOULI***	***:Professeur***	***à l'Université de Carthage***	***:Président***
♦ ***M. Zied ELOUEDI***	***:Maître de conférences***	***à l'Université de Tunis***	***:Directeur de thèse***
♦ ***M. Pawan LINGRAS***	***:Professeur***	***à l'Université de Saint Mary's, Canada***	***:Co-directeur de thèse***
♦ ***M. Abdelwahed TRABELSI***	***:Professeur***	***à l'Université de Tunis***	***:Rapporteur***
♦ ***Mme Nahla BEN AMOR***	***:Maître de conférences***	***à l'Université de Tunis***	***:Rapporteur***
♦ ***M. Mohamed HAOUARI***	***:Professeur***	***à l'Université de Tunis***	***:Membre***

Thèse / ISG.Tunis / Septembre 2011

Belief Rough Set Classifier

Salsabil TRABELSI

Acknowledgements

There are so many people who have provided me with their support during the achievement of my PhD. First, I would like to thank my advisor Dr. Zied Elouedi for his continuous guidance and for all the valuable discussions we have had during the period of work. I thank him for his support and for his constant encouragements.

I would like to express my sincere gratitude to my co-advisor Pr. Pawan Lingras in the fulfillment of my PhD. He has been very patient in supervising this work at all its stages. His competence and his generosity have allowed me to work in ideal conditions. His comments and his observations have always been of great interest.

I am also thankful to Dr. Richard Jensen from *University of Aberystwyth, UK* for replying me each time I asked him questions regarding rough set theory.

My thanks go to all members of my laboratory LARODEC especially Pr. Khaled Mellouli (University of Carthage, Tunisia) who introduced me, with Dr. Zied Elouedi, to the area of research and especially to the area of uncertainty and data mining. Thank you for your constant support.

Finally, I want to express my thanks and love to my husband, my son, my parents, my parents in law, my brothers and sisters, my brothers and sisters in law and to all the persons who love me. This work would not have been possible without your support.

Contents

List of Figures

List of Tables

Introduction

Classification refers to data mining problems of attempting to predict the category of unseen objects by building a model based on some predictor variables. There is a number of classification techniques which are widely used in artificial intelligence and machine learning like decision trees, artificial neural networks, k-nearest neighbors, naive bayesian networks, etc. These techniques are applied in several real-world applications such as marketing, medical diagnosis, credit approval, etc.

The rough set theory based on approximation reasoning and proposed by Pawlak (1982, 1991) also constitutes a sound basis for data mining applications and knowledge discovery. This theory provides efficient methods, algorithms and tools for finding hidden patterns in databases. It also allows for attribute reduction by finding minimal sets of data with the same knowledge as in the original data. It can also be used to evaluate the significance attributes and to identify partial or total dependencies in databases. The rough set approach offers solutions to the problem of feature selection, discretization, data reduction and decision rule generation to classify new objects, etc.

Classification technique based on rough sets called *Rough Set Classifier* (RSC) is a successful classification technique applied in several real-world applications such as medicine, finance, telecommunication, intelligent agents, image analysis, pattern recognition, marketing, etc. The models generated by rough set classifier take the form of 'IF-THEN' rules. RSC performs feature selection and data reduction before generating rules, this method can avoid many iterations. It is an efficient technique that can produce optimal and minimal set of decision rules. RSC has the advantages of time complexity of learning, accuracy and size of the discovered rules. It is not true for some other classification techniques which have the problems of size, accuracy and time complexity of learning models especially from large databases

(Trabelsi et al., 2006, 2007).

The standard rough set classifier like the other classification techniques, do not perform their classification task very well in an environment characterized by uncertainty or incomplete data. Many researchers have extended rough sets and its applications to accommodate uncertainty (Grzymala-Busse, 2003, 2004; Hong et al., 2002; Kryszkiewicz, 1995). These extensions of rough set classifiers work with incomplete or missing data sets. Two main cases of missing condition attribute values are considered: 'lost' (the original value was erased) and 'do not care' conditions (the original value was irrelevant). These extensions do not deal with partially uncertain condition or decision attribute values in decision system. This kind of uncertainty exists in many real-world applications like in medicine where diseases (decision attribute) of some patients (objects) or even their symptoms (condition attributes) may be partially uncertain. It is not useful to eliminate these objects from classification process because it represents a loss of important information.

In this work, we will focus on rough set classifiers due to its advantages and we will extend it to deal with partial uncertainty. The latter appears only in decision attribute values of decision system. We handle only symbolic condition and decision attribute values. To deal with this kind of uncertainty, we will choose the belief function theory which is able to represent the partial or the total ignorance in a flexible way. It is considered to be a useful theory for representing and managing uncertain knowledge. The belief function theory is appropriate to handle uncertainty in classification problems such as belief decision tree (Denœux & Skarstein-Bjanger, 2000; Elouedi et al., 2001; Vannoorenberghe & Denœux, 2002), belief clustering (Ben-Hariz et al., 2006; Masson & Denœux, 2004), belief neural networks (Denœux, 2000) and belief K-nearest neighbors (Denœux, 1995).

Our thesis deals with the problem of learning decision rules based on rough set methodology from dataset characterized by uncertain decision attribute values. The uncertainty is represented by the *Transferable belief Model* (TBM), one interpretation of the belief function theory. To solve this problem, we propose two classification systems based on rough set methodology. The first classification technique is called *Belief Rough Set Classifier* (BRSC) which is based on the new definition of the basic concepts of rough sets under belief function framework. The second is more sophisticated and is called *Belief Rough Set Classifier* based on *Generalization Distribution Ta-*

ble (BRSC-GDT) which is derived from the hybrid system named GDT-RS. The latter is a combination of *Generalization Distribution Table* (GDT) and *Rough Sets* (RS) which is also redefined under the belief function framework to be called belief GDT-RS. Our solutions aim at generating from uncertain data a minimal and a significant set of decision rules to classify unseen instances.

To improve the time requirement of learning models relative to both the classifiers: BRSC and BRSC-GDT, we propose a heuristic feature selection method based on rough sets under the uncertain context. By applying it in a pre-processing stage, we obtain two new versions of BRSC and BRSC-GDT denoted respectively by H-BRSC and H-BRSC-GDT. Furthermore, to improve the classification power, we propose a dynamic reduct method which yields to more stable results under the uncertain context. The latter notion yields to new versions of BRSC and BRSC-GDT denoted respectively by D-BRSC and D-BRSC-GDT.

Note that there is also another similar classifier denoted *Belief Decision Tree* (BDT) (Elouedi et al., 2001) which induces uncertain decision rules from uncertain databases that are able to classify new instances. The uncertainty also appears in decision attributes values and is represented through the TBM. There are two approaches of building the BDT: the averaging and the conjunctive approaches. However, inducing a BDT may lead in most cases to very large trees with bad classification accuracy and difficult comprehension. To cope with this problem, pre-pruning (Elouedi et al., 2002) and post-pruning (Trabelsi et al., 2007) methods have been applied to simplify the belief decision tree and improve its classification accuracy. Hence, the BDT needs the step of pruning which is not a simple task. Our two new solutions based on rough set theory generate a minimal and an efficient set of decision rules without costly calculation, without many iterations. We hope to obtain satisfactory results without creating a decision tree and without pruning it.

To show the applicability and the effectiveness of our two classification techniques based on rough sets with their versions, we carry experimentations on modified real-world databases to include uncertainty and on a naturally uncertain web usage mining database. Three evaluation criteria are chosen : time requirement of learning, size of models and classification accuracy. Then, we compare the results with those obtained from the *Belief Decision Tree* (BDT), which also deals with uncertain decision attribute.

Our thesis is organized as follows:

- *Part I* presents the necessary theoretical aspects concerning the basic notions of belief function theory as understood through TBM. In addition, this part also details the basic concepts of rough set theory and rough set classifier.

- *Part II* details the proposed developments that we have made to build our two classification approaches based on rough sets namely *Belief Rough Set Classifier* and *Belief Rough Set Classifier* based on *Generalization Distribution Table*. The knowledge about decision attribute values of the training objects is represented by belief function theory. In this part, the main basic concepts of our classification systems are detailed like the new definitions of the basic concepts of rough sets and the hybrid system GDT-RS under the belief function framework. Then, we have detailed the two main procedures needed to create them namely construction and classification procedures. Next, we have proposed two ideas to improve the results relative to the construction procedure of our classifiers like the use of a heuristic and a dynamic feature selection methods. Finally, to judge the performance of our two proposed classifiers and their versions, experimentations on modified real-world databases are made where the uncertainty in the decision attribute is created artificially and on a naturally uncertain web usage database obtained from web access logs of the introductory computing science course at Saint Mary's University. The uncertainty appears only in decision attributes and is handled by the belief function theory.

Finally, a general conclusion summarizes the major achievements of this thesis and presents possible future developments. Two appendices complete this thesis. The first appendix details the major programs and algorithms needed to build and test our two approaches which are implemented to check their feasibility and qualities. The second appendix gives a part of the naturally uncertain web usage mining database used in the experimental part.

Part I

Theoretical Aspects

Our thesis deals with the elaboration of two new classification approaches based on rough set theory called *Belief Rough Set Classifier* (BRSC) and *Belief Rough Set Classifier* based on *Generalization Distribution Table* (BRSC-GDT). These proposed models make it possible to learn decision rules from uncertain data where the uncertainty is represented by the belief function theory. Hence, in this first part of our thesis, we present the theoretical aspects which are useful to understand the developments made in this thesis concerning the BRSC and the BRSC-GDT.

This part presents the necessary background concerning the belief function theory and the rough set theory. Hence, it is composed of two chapters:

- The first chapter deals with the belief function theory as understood in the transferable belief model. This theory is a useful tool which offers a convenient framework thanks to its ability to represent uncertain knowledge.

  This chapter presents the basic concepts of this theory. Next, some special belief functions are described. Finally, useful concepts within the belief function theory are detailed namely the combination, the conditioning, the discounting and also the decision process. The different notions are illustrated by examples.

- The second chapter presents the rough set theory as a new mathematical approach to deal with imperfect knowledge. Rough sets have been proposed for a very wide variety of applications. In particular, the rough set approach seems to be important for machine learning and knowledge discovery.

  This chapter details the basic concepts of the rough set theory such as decision table, indiscernibility relation, set approximations, positive region and reduct. The rough set theory offers solution for *Knowledge Discovery Database* (KDD) process such as discretization, data reduction, feature selection and generation of decision rules. In this chapter, the classification process based on rough sets, called *Rough Set Classifier* (RSC), is also presented with its two main procedures. Some variants and extensions of rough set classifier are presented like combination with other theories of uncertainty to handle incomplete data. Illustrative examples are also provided in order to further explain various notions.

Chapter 1

Belief function theory

1.1 Introduction

The theory of belief functions has been proposed for modeling someone's degrees of belief to quantify subjective judgments. It is considered as a useful theory for representing and managing uncertain knowledge because of its relative flexibility. This theory was introduced by Dempster (1968) and Shafer (1976). Hence, it is usually called the Dempster-Shafer theory. The belief function theory is widely applied in artificial intelligence and provides sound and elegant solutions to real life problems. There are several interpretations of this theory: the lower probability model (Walley, 1991), the Dempster's model (Dempster, 1967, 1968), the theory of hints (Kohlas & Monney, 1995) and the *Transferable Belief Model* (TBM) (Smets, 1988, 1998b; Smets & Kennes, 1994; Smets & Kruse, 1997). Each model corresponds to different understandings of the concept of uncertainty.

TBM is not an adaptation of probability theory. It seems to fit essentially with what Shafer (1976) developed in his work. It is developed to quantify beliefs. It covers the same domain as the Bayesian-subjectivist probabilities except it is based on belief functions (including Dempster's rule of conditioning and Dempster's rule of combination). In this thesis, we focus only on the TBM. Consequently, the presentation in this chapter will be based on this model. This chapter presents an overview of some basic concepts of the belief function theory. Next, useful concepts are also detailed like the combination, the conditioning, the discounting and also the decision process. The different notions are illustrated by examples.

1.2 Basic concepts

1.2.1 Frame of discernment

Let Θ be a finite set of elementary events to a given problem, called the frame of discernment. These events are assumed to be exhaustive and mutually exclusive. Such set Θ is also referred to as the universe of discourse or the domain of reference (Smets, 1988). All the subsets of Θ belong to the power set of Θ, denoted by 2^Θ and defined as follows:

$$2^\Theta = \{E : E \subseteq \Theta\} \tag{1.1}$$

Every element of 2^Θ is called a proposition or an event. It can also be seen as a possible answer to a given question. Note that the empty set $\emptyset$ belongs to the power set 2^Θ and it corresponds to the impossible proposition (the contradiction), whereas the set Θ corresponds to the certain proposition (the tautology).

Example 1.1 *Let us treat the problem of identification 'Who murdered John?' and we have three suspects. Thus, the frame of discernment related to this problem is defined as follows:*

$\Theta = \{Henry,\ Peter,\ Sara\}$

The power set of Θ *is:*
$2^\Theta = \{\emptyset, \{Henry\}, \{Peter\}, \{Sara\}, \{Henry,\ Peter\}, \{Peter,\ Sara\}, \{Henry,\ Sara\}, \{Henry,\ Peter,\ Sara\}\}$

1.2.2 Basic belief assignment

The impact of a piece of evidence on the different subsets of the frame of discernment Θ is represented by *basic belief assignment* (bba), called initially by Shafer (1976) *basic probability assignment* (bpa), an expression that has unfortunately created serious confusion in the past. The bba is defined as follows:

$$\mathrm{m}: 2^\Theta \rightarrow [0,1]$$

$$\sum_{E \subseteq \Theta} m(E) = 1 \tag{1.2}$$

The value *m(E)*, named a *basic belief mass* (bbm), represents the portion of belief committed exactly to the event E. Due to the lack of information, this quantity cannot be apportioned to any strict subset of E. So, it represents the direct specific support of evidence on E.

The quantity $m(\Theta)$ measures the portion of belief assigned to the whole frame Θ. It represents the beliefs not assigned to the different subsets of Θ.

Shafer (1976) initially proposed a normality condition expressed by:

$$m(\emptyset) = 0 \tag{1.3}$$

Such bba is called a normalized basic belief assignment.

Smets (1990) relaxed this condition and interpreted $m(\emptyset)$ as the amount of conflict between the pieces of evidence or as the part of belief given to the fact that none of the hypotheses in Θ is true, in other words the hypotheses making up the frame of discernment of hypotheses are not exhaustive. This last interpretation refers to the so-called open-world assumption (Smets, 1990) (in contrast to the exhaustive frame of discernment which is referred to as the closed-world assumption).

Example 1.2 *Assume* $\Theta = \{$*Henry, Peter, Sara*$\}$.

The bba related to a piece of evidence concerning the murderer of John is defined as follows:

m($\{$*Henry*$\}$*)=0.1;*
m($\{$*Henry, Peter*$\}$*)=0.7;*
m($\{$*Sara*$\}$*)=0.2;*

For example, 0.1 represents the part of belief exactly supporting that the murderer is Henry.

1.2.3 Belief function

A belief function, denoted *bel*, quantifies the total amount of $justified$ $specific$ support given to E. The belief function *bel* (Shafer, 1976) is defined as follows:

$$bel(E) = \sum_{\emptyset \neq F \subseteq E} m(F) \tag{1.4}$$

The belief is $justified$ because $bel(E)$ includes all the basic belief masses given to the subsets of E, in contrast to the bba which expresses only the part of belief committed exactly to E. The belief function is $specific$ because $bel(E)$ includes only the bbm given to the subsets that support E without supporting $\bar{E}$. The basic belief mass $m(\emptyset)$ is not included in $bel(E)$ as it is given to the subset $\emptyset$ that supports not only E but also its complement $\bar{E}$.

Properties

- The total belief committed to the empty set is defined as follows:

$$bel(\emptyset) = 0 \tag{1.5}$$

  Usually, $bel(\Theta)$=1 is assumed (Shafer, 1976) corresponding to a closed-world assumption. It can be ignored for the open-world assumption and we only require $bel(\Theta) \leq 1$.

- $m(E)$ may be expressed by the values of bel as follows (Smets, 2002):

$$m(E) = \sum_{F \subseteq E} (-1)^{|E|-|F|} bel(F), \forall E \subseteq \Theta, E \neq \emptyset \tag{1.6}$$

- The bbm $m(\emptyset)$ is computed as follows:

$$m(\emptyset) = 1 - bel(\Theta) \tag{1.7}$$

- The TBM departs from Bayesian approach based on belief functions in that the additivity encountered in probability theory is not assumed. It is replaced by inequalities:

$$bel(E \cup F) \geq bel(E) + bel(F) - bel(E \cap F), For\, E, F \subseteq \Theta \tag{1.8}$$

- Sub-additivity:

$$bel(E) + bel(\bar{\mathrm{E}}) \leq 1 \tag{1.9}$$

  This rule shows that the knowledge of the belief given to a proposition E does not necessarily give us an information about the degree of belief of the proposition $\bar{E}$.

  Contrary to the theory of probability, in the belief function theory increasing beliefs on a proposition E does not necessary require the decrease of beliefs on $\bar{E}$.

- Monotonicity:

$$E \subseteq F \implies bel(F) \geq bel(E) \tag{1.10}$$

  Hence, Θ will get the highest value of bel (the upper limit), whereas $\emptyset$ will get the lowest value (the lower limit).

Example 1.3 *The belief function bel corresponding to the bba m (see example 1.2) is defined as follows:*

bel($\emptyset$)=0;
bel({Henry})=0.1;
bel({Peter})=0;
bel({Sara})=0.2;
bel({Peter, Sara})=0.7;
bel({Henry, Peter})=0.1+0.7=0.8;
bel({Henry, Sara})=0.1+0.2=0.3;
bel(Θ)=0.1+0.7+0.2=1;

For example, 0.8 is the total belief committed to the proposition {Henry, Peter}.

1.2.4 Plausibility function

The plausibility function *pl* quantifies the maximum amount of *potential specific* support of belief that could be given to a proposition E of the frame of discernment. It is obtained by adding all those basic belief masses given to propositions F compatible with E.

The word *potential* is used because the basic belief masses included in $pl(E)$ could be transfered to non-empty subsets of E if new information could justify such a transfer.

The plausibility function *pl* is defined as follows:

$$pl(E) = \sum_{E \cap F \neq \emptyset} m(F) \tag{1.11}$$

$$= bel(\Theta) - bel(\bar{\mathrm{E}}) \tag{1.12}$$

$$= \sum_{F \subseteq \Theta} m(F) - \sum_{F \subseteq \bar{\mathrm{E}}} m(F) \tag{1.13}$$

Properties

- For the closed-world and the open-world assumptions, we have

$$bel(\emptyset) = pl(\emptyset) = 0 \tag{1.14}$$

  Note that under open-world assumption, the basic belief mass $m(\emptyset)$ should not be included in $bel(E)$ nor in $pl(E)$, as it is given to the subset $\emptyset$ that supports not only E but also $\bar{\mathrm{E}}$. This is the origin of $specific$ support.

- For the open-world assumption, we have

$$bel(\Theta) = pl(\Theta) = 1 - m(\emptyset) \leq 1 \tag{1.15}$$

- For the closed-world assumption, we have

$$bel(\Theta) = pl(\Theta) = 1 \tag{1.16}$$

- Over-additivity:

$$pl(E) + pl(\bar{\mathrm{E}}) \geq 1 \tag{1.17}$$

- Monotonicity:

$$E \subseteq F \implies pl(F) \geq pl(E) \tag{1.18}$$

- $pl(E)$ can be expressed by $bel(E)$, we get:

$$pl(E) = bel(E) + \sum_{E \cap F \neq \emptyset, F \not\subseteq E} m(F) \tag{1.19}$$

- For E, F $\subseteq \Theta$, E $\cap$ F $= \emptyset$,

$$pl(E \cup F) \leq pl(E) + pl(F) \tag{1.20}$$

- For E $\subseteq \Theta$,

$$bel(E) \leq pl(E) \tag{1.21}$$

Example 1.4 *The plausibility function pl corresponding to the bba m (see example 1.2) is defined as follows:*

$pl(\emptyset)=0;$
$pl(\{Henry\})=0.1+0.7=0.8;$
$pl(\{Peter\})=0.7;$
$pl(\{Sara\})=0.2;$
$pl(\{Henry, Peter\})=0.1+0.7=0.8;$
$pl(\{Henry, Sara\})=0.1+0.7+0.2=1;$
$pl(\{Peter, Sara\})=0.7+0.2=0.9;$
$pl(\Theta)=0.1+0.7+0.2=1;$

For example, 0.2 represents the maximum degree of belief that the proposition $\{Sara\}$ may have.

1.2.5 Commonality function

Another function is basically used to simplify computations in the belief function theory namely the commonality function q. It has no intuitive interpretation. However, it may represent the total mass that is free to move to every element of E (Barnett, 1991). It is defined as follows:

$$q(E) = \sum_{E \subseteq F} m(F) \tag{1.22}$$

Properties

- The commonality value relative to the empty set is defined as follows:

$$q(\emptyset) = 1 \tag{1.23}$$

- The commonality value relative to the whole frame of discernment is defined as follows:

$$q(\Theta) = m(\Theta) \tag{1.24}$$

Example 1.5 *The commonality function q corresponding to the bba m (see example 1.2) is defined as follows:*

$q(\emptyset)=1;$
$q(\{Henry\})=0.1+0.7=0.8;$
$q(\{Peter\})=0.7;$
$q(\{Sara\})=0.2;$

q({Henry, Peter})=0.7;
q({Henry, Sara})=0;
q({Peter, Sara})=0;
q(Θ)=0;

Remark:

The basic belief assignment (m), the belief function (bel), the plausibility function (pl) and the commonality function (q) are considered as different expressions of the same information (Denœux, 1999).

1.2.6 Focal elements, body of evidence and core

The subsets E of the frame of discernment Θ such that *m(E)* is strictly positive are the focal elements of the bba *m*.

The pair (Fc, m) is called a body of evidence where Fc is the set of all the focal elements relative to the bba *m*.

The union of all the focal elements of m are named the core and are defined as follows:

$$\varphi = \bigcup_{E:m(E)>0} E \tag{1.25}$$

Example 1.6 *Let us continue with the example 1.2, the subsets {Henry}, {Henry, Peter}, and {Sara} are the focal elements of the bba m.*

So, (Fc, m) is called the body of evidence such that: F={{Henry}, {Henry, Peter}, {Sara}}

The core of this m is defined as follows:
$\varphi = \{Henry\} \cup \{Henry,\ Peter\} \cup \{Sara\} = \Theta$

1.3 Special belief functions

In the literature, several kinds of belief functions are proposed. Such functions are used to express particular situations related generally to uncertainty.

1.3.1 Vacuous belief function

A vacuous belief function is a normalized belief function defined such that (Shafer, 1976):

$$m(\Theta) = 1 \; and \; m(E) = 0 \, for \, E \neq \Theta \tag{1.26}$$

The advantage of the TBM over the classical Bayesian approach resides in its large flexibility, its ability to represent every state of partial beliefs, up to the state of total ignorance by the vacuous belief function.

Example 1.7 *Assume an expert was not able to detect the murderer. Hence, we get a state of total ignorance where the corresponding bba is defined as follows:*

m(Θ)=1 and m(E)=0 for $E \neq \Theta$

1.3.2 Categorical belief function

A categorical belief function is a normalized belief function such that its bba is defined as follows (Mellouli, 1987):

$$m(E) = 1 \, for \, some \, E \subset \Theta \; and \; m(F) = 0, \, for F \subseteq \Theta, F \neq E \tag{1.27}$$

Example 1.8 *Assume an expert was certain that 'the murderer is a male'. So, the corresponding bba presents a categorical belief function defined as follows:*

m({Henry, Peter})=1;

1.3.3 Certain belief function

A certain belief function is a categorical belief function such that its focal element is a singleton. Its corresponding bba is defined as follows:

$$m(E) = 1 \; and \; m(F) = 0 \, for \, all \, F \neq E \, and \; F \subseteq \Theta \tag{1.28}$$

where E is a singleton event of Θ.
Such function represents a state of total certainty as it assigns all the belief to a unique elementary event.

Example 1.9 *Assume an expert affirms that the murderer is a woman. So, the corresponding bba presents a certain belief function defined as follows:*

m({Sara})=1;

1.3.4 Simple support function

A belief function is called a *simple support function* (ssf) if it has at most one focal element different from the frame of discernment Θ. This focal element is called the focus of the ssf. A simple support function is defined as follows (Smets, 1995):

$$m(X) = \begin{cases} w, & if\ X = \Theta \\ \text{1-w}, & if\ X = E\ for\ some\ E \subseteq \Theta \\ 0, & otherwise. \end{cases} \tag{1.29}$$

Where E is the focus and $w \in [0,1]$.

This ssf describes a belief function induced by a piece of evidence supporting E (with $1 - w$) and leaving the remaining beliefs for Θ.

Note that the empty set $\emptyset$ can be considered as a focus of a simple support function.

Example 1.10 *Let us continue with the example 1.1. Assume we have a bba defined as follows:*

m({Sara, Peter})=0.7;
m(Θ)=0.3;

m is called a simple support function where the focus is the proposition {Sara, Peter}.

1.3.5 Bayesian belief function

A belief function is said to be Bayesian if its focal elements are singletons (Shafer, 1976). Hence, *bel* becomes a probability distribution. It is defined as follows:

$$bel(\emptyset) = 0 \tag{1.30}$$

$$bel(\Theta) = 1 \tag{1.31}$$

$$bel(E \cup F) = bel(E) + bel(F)\ Where\ E,\, F\ \subset \Theta\, and\, E \cap F = \emptyset \tag{1.32}$$

Properties

- *bel* is a Bayesian belief function if all its focal elements are singletons. Hence, *bel* becomes a probability distribution.

- As in the probability theory:

$$bel(E) + bel(\bar{\mathrm{E}}) = 1 \, for \, E \subset \Theta \tag{1.33}$$

- In the case of Bayesian belief functions we get:

$$bel = pl \tag{1.34}$$

Example 1.11 *Let us consider* Θ=\{*Henry, Peter, Sara*\}.

We get a piece of evidence expressed by the following bba m:

m(\{Henry\})=0.4;
m(\{Peter\})=0.5;
m(\{Sara\})=0.1;
m(Θ)=0;

The bba m is a Bayesian bba since all its focal elements are singletons.

1.3.6 Consonant belief function

A belief function is said to be consonant if its focal elements (E_1, E_2, ...,E_n) are nested, that is $E_1 \subseteq E_2 \subseteq .... \subseteq E_n$.

Properties

- Every simple support function is a consonant belief function.
- *bel* is a necessity measure (Smets, 1995):

$$bel(E \cap F) = min(bel(E), bel(F)) \tag{1.35}$$

- *pl* is a possibility measure (Smets, 1995):

$$pl(E \cup F) = max(pl(E), pl(F)) \tag{1.36}$$

Example 1.12 *Let us consider this bba defined as follows:*

m(\{Henry\})=0.2;
m(\{Henry, Peter\})=0.5;
m(Θ)=0.3;

The focal elements of this bba m are nested. Hence, it is a consonant bba.

1.3.7 Dogmatic and non-dogmatic belief functions

A belief function is said to be dogmatic if and only if its corresponding bba m is such that $m(\Theta) = 0$. This case involves some previous cases (certain belief functions, categorical belief functions). A non-dogmatic belief function is defined such that $m(\Theta) > 0$ (Smets, 1995).

1.4 Combination

The belief function theory, as understood in the TBM framework, offers interesting rules for aggregating the basic belief assignments (bba's) induced from distinct pieces of evidence and provided by two (or more) sources of information.

Let m_1 and m_2 be two bba's defined on the same frame of discernment Θ. These two bba's are collected by two distinct pieces of evidence and induced from two experts (information sources). These bba's can be combined either conjunctively or disjunctively.

1.4.1 Conjunctive rule of combination

When we know that both sources of information are fully reliable then the bba representing the combined evidence satisfies (Smets, 1998a):

$$(m_1 \bigcirc\!\!\!\!\!\cap\, m_2)(E) = \sum_{F,G \subseteq \Theta : F \cap G = E} m_1(F) m_2(G) \tag{1.37}$$

This rule can be simply computed in terms of the commonality functions as follows:

$$(q_1 \bigcirc\!\!\!\!\!\cap\, q_2)(E) = q_1(E) q_2(E) \tag{1.38}$$

where q_1 and q_2 are respectively the commonality functions corresponding respectively to the bba's m_1 and m_2. This implies the most useful relation that explains the usefulness of the commonality function.

The conjunctive rule is considered as unnormalized Demspter's rule of combination dealing with the closed-world assumptions, defined as follows (Shafer, 1976, 1986):

$$(m_1 \oplus m_2)(E) = K (m_1 \bigcirc\!\!\!\!\!\cap\, m_2)(E) \tag{1.39}$$

Where

$$K^{-1} = 1 - (m_1 \bigcirc\!\!\!\!\cap\, m_2)(\emptyset) \tag{1.40}$$

$$and \ (m_1 \oplus m_2)(\emptyset) = 0 \tag{1.41}$$

K is called *the normalization factor.*

Properties

The conjunctive rule of combination is characterized by the following properties:

- Compositionality:

  $(m_1 \bigcirc\!\!\!\!\cap\, m_2)$(E) is function of E, m_1 and m_2.

- Commutativity:

$$m_1 \bigcirc\!\!\!\!\cap\, m_2 = m_2 \bigcirc\!\!\!\!\cap\, m_1 \tag{1.42}$$

- Associativity:

$$(m_1 \bigcirc\!\!\!\!\cap\, m_2) \bigcirc\!\!\!\!\cap\, m_3 = m_1 \bigcirc\!\!\!\!\cap\, (m_2 \bigcirc\!\!\!\!\cap\, m_3) \tag{1.43}$$

  The result of the combination of pieces of evidence is independent of the order in which they are considered and they are associated.

- Non-idempotency:
  The conjunctive rule of combination is not idempotent. So,

$$(m \bigcirc\!\!\!\!\cap\, m) \neq m \tag{1.44}$$

- Neutral element:
  The neutral element within the conjunctive rule of combination is the vacuous basic belief assignment representing the total ignorance.

$$(m \bigcirc\!\!\!\!\cap\, m_0) = m \tag{1.45}$$

  Where m_0 is a vacuous bba.

Example 1.13 *Assume* Θ=\{*Henry, Peter, Sara*\}.

Let two bba's m_1 *and* m_2 *relative to two pieces of evidence.*

$m_1(\{Henry\})=0.6;$
$m_1(\{Henry, Peter\})=0.2;$
$m_1(\Theta)=0.2;$

$m_2(\{Peter\})=0.4;$
$m_2(\{Henry, Peter\})=0.3;$
$m_2(\Theta)=0.3;$

Applying the conjunctive rule of combination, we get:

$(m_1 \textcircled{\cap} m_2)(\emptyset)=0.24;$
$(m_1 \textcircled{\cap} m_2)(\{Henry\})=0.36;$
$(m_1 \textcircled{\cap} m_2)(\{Peter\})=0.16;$
$(m_1 \textcircled{\cap} m_2)(\{Henry, Peter\})=0.18;$
$(m_1 \textcircled{\cap} m_2)(\Theta)=0.06;$

1.4.2 Disjunctive rule of combination

The dual of the conjunctive rule of combination that builds the bba representing the impact of two pieces of evidence when we only know that at least one of sources of information is reliable but we do not know which is reliable, then the bba representing the combined evidence satisfies (Smets, 1998a):

$$(m_1 \textcircled{\cup} m_2)(E) = \sum_{F,G \subseteq \Theta : F \cup G = E} m_1(F) m_2(G) \tag{1.46}$$

The disjunctive rule of combination, as the conjunctive rule of combination, is commutative and associative.

- Commutativity:
$$m_1 \textcircled{\cup} m_2(E) = m_2 \textcircled{\cup} m_1(E) \tag{1.47}$$
- Associativity:
$$(m_1 \textcircled{\cup} m_2(E)) \textcircled{\cup} m_3(E) = m_1 \textcircled{\cup} (m_2(E) \textcircled{\cup} m_3(E)) \tag{1.48}$$

Example 1.14 *Assume* Θ=\{*Henry, Peter, Sara*\}.

Let two bba's m_1 *and* m_2 *relative to two pieces of evidence see in example 1.14. By applying the disjunctive rule of combination, we get:*

$(m_1 \bigcirc\!\!\!\!\cup\, m_2)(\{Henry, Peter\})=0.56;$
$(m_1 \bigcirc\!\!\!\!\cup\, m_2)(\Theta)=0.44;$

Remark:

Since the conjunctive and the disjunctive rules of combination are both commutative and associative, combining several pieces of evidence induced from distinct information sources (either conjunctively or disjunctively) may be easily ensured by applying repeatedly the chosen rule.

1.5 Conditioning

Suppose you have some belief on Θ represented by the basic belief assignment m. Then, some further evidence becomes available to you and this piece of information implies that the actual world cannot be in one of the worlds in $\bar{\mathrm{F}}$. Then, the mass $m(E)$ that initially was supporting that the actual world is in E now supports that the actual world is in $F \cap E$ as every world in $\bar{\mathrm{F}}$ must be 'eliminated'. So, $m(E)$ is transferred to $F \cap E$ after conditioning on F. The TBM gets its name from this transfer operation.

This operation leads to the conditional basic belief assignment $m[F]$, belief function $bel[F]$ and plausibility function $pl[F]$, defined as follows:

$$m[F](E) = \sum_{G \subseteq \bar{\mathrm{F}}} m(E \cap G) \tag{1.49}$$

$$bel[F](E) = bel(E \cup \bar{\mathrm{F}}) - bel(\bar{\mathrm{F}}) \tag{1.50}$$

$$pl[F](E) = pl(E \cap F) \tag{1.51}$$

$$q[F](E) = \begin{cases} q(E), & \text{if E} \subseteq \text{F} \\ 0, & \text{otherwise.} \end{cases} \tag{1.52}$$

The rule by which these equations are built is called Dempster's rule of conditioning (Smets, 1988).

Example 1.15 *Let us take the same bba m concerning the murderer of John see in example 1.2. If you learn that the killer not Peter. Hence, the mass m({Henry, Peter})=0.7 should be eliminated. So, there are a mass transfer to:*

m({Henry})=0.1+ 0.7=0.8;
m({Sara})=0.2;

1.6 Discounting

Dealing with evidence expressed by experts requires to take into account the level of expertise of each information source. In fact, experts are not fully reliable and a method of discounting seems imperative to update experts' beliefs by taking into account their reliability. The idea is to weight most heavily the opinions of the best experts and conversely for the less reliable ones.

In the transferable belief model, discounting allows to take in consideration the reliability of the information source that generates the bba m. For $\alpha \in [0,1]$, let (1-α) be the degree of confidence ('reliability') we assign to the source of information. If the source is not fully reliable, the bba it generates is 'discounted' into a new less informative bba denoted m^α (Smets, 1992):

$$m^\alpha(E) = (1-\alpha).m(E), \qquad for\, E \subset \Theta \tag{1.53}$$

$$m^\alpha(\Theta) = \alpha + (1-\alpha).m(\Theta) \tag{1.54}$$

Properties

- $\alpha = 0$ means that expert is totally reliable.
- $\alpha = 1$ means that the expert is not reliable at all. His opinions have to be totally ignored.

Example 1.16 *Let us discount the bba m (given in the example 1.2) with degree of reliability* $(1-\alpha)$ *is equal to 0.9, so we get:*

m^α*({Henry})=0.9*0.1=0.09;*
m^α*({Henry, Peter})=0.9*0.7=0.63;*
m^α*({Sara})=0.9*0.2=0.18;*
$m^\alpha(\Theta)$*=0.1*;

1.7 Decision process

The theory of belief functions is characterized by its ability to handle uncertainty and to ensure the combination of evidence induced from different sources. In order to make a decision, we hope to select the most likely hypothesis which may be difficult to realize directly with the basics of the belief function theory where bbm's are given not only to singletons but also to subsets of hypotheses.

In this section, we present some solutions allowing to ensure the decision making within the belief function theory. The best known is the pignistic probability proposed by the transferable belief model (Smets, 1988, 1998b; Smets & Kennes, 1994; Smets & Kruse, 1997). Other criteria will be presented like the maximum of credibility and the maximum of plausibility (Janez, 1996).

1.7.1 Pignistic probability

In the transferable belief model, holding beliefs and making decisions are distinct processes. Hence, it proposes two levels:

- *The credal level* where beliefs are entertained and represented by belief functions.
- *The pignistic level* where beliefs are used to make decisions and represented by probability functions called the pignistic probabilities and is defined as:

$$BetP(\{a\}) = \sum_{F \subseteq \Theta} \frac{\mid \{a\} \cap F \mid}{\mid F \mid} \frac{m(F)}{(1 - m(\emptyset))}, \; for\, all\, a \in \Theta \tag{1.55}$$

Example 1.17 *To make a decision, we have to compute the pignistic probability BetP corresponding to the bba m (given in the example 1.2), we get:*

BetP({Henry})=0.45;
BetP({Peter})=0.35;
BetP({Sara})=0.2;

It is more probable that the murderer is Henry.

1.7.2 Maximum of credibility

It consists in choosing the hypothesis having the highest value of the belief function *bel*, that is the most credible hypothesis.

Decision based on the maximum of credibility is considered as a pessimistic approach since it chooses the 'best' hypothesis based on the minimum 'chance' to realize.

Example 1.18 *Let us continue with the example 1.2, and try to make a decision by using the criterion of the maximum of credibility. We get:*

bel({Henry})=0.1;
bel({Peter})=0;
bel({Sara})=0.2;

The largest value of bel is the one assigned to the hypothesis {Sara}. According to the maximum of credibility, Sara is the murderer.

1.7.3 Maximum of plausibility

It consists in choosing the hypothesis having the highest value of the plausibility function *pl* which means that we support the hypothesis that gives the less evidence for the contrary hypothesis.

Contrary to the maximum credibility criterion, this criterion is considered as optimistic since it takes into account of the maximum of 'chance' of realization of each hypothesis.

Example 1.19 *Let us continue with the example 1.2, and try to make a decision by using the criterion of the maximum of credibility. We get:*

pl({Henry})=0.8;
pl({Peter})=0.7;
pl({Sara})=0.2;

The largest value of pl is the one assigned to the hypothesis {Henry}. So, according to the maximum of plausibility, Henry is the murderer.

1.8 Conclusion

In this chapter, we have presented the basic notions of belief function theory as understood in the transferable belief model using illustrative examples. This theory is appropriate to handle uncertainty in classification problems (Ben-Hariz et al., 2006; Denœux, 1995, 2000; Denœux & Skarstein-Bjanger, 2000; Elouedi et al., 2001).

Let us remember that the objective of our thesis is to learn decision rules using classification systems based on rough set theory, called *Belief Rough Set Classifier* (BRSC) and *Belief Rough Set Classifier* based on *Generalization Distribution Table* (BRSC-GDT), from uncertain data where the uncertainty is handled through the TBM. Hence, the following chapter will deal with the theory of rough sets where its basics will be described.

Chapter 2

Rough set theory

2.1 Introduction

The idea of *Rough Sets* (RS) was proposed by Pawlak (1982, 1991) as a new mathematical tool to deal with vague concepts. The main goal of the rough set analysis is the induction of approximations of concepts. Rough set theory constitutes a sound basis for KDD as a tool to discover hidden patterns from data. It offers solutions to the problem of decision table simplification which yields fast computer algorithms, discretization, and decision rule generation, etc. The theory of rough sets has been followed by the development of several software systems that implement rough set operations.

Classification approach based on rough set theory, denoted *Rough Set Classifier* (RSC), is a successful classification technique applied in several real-world applications (An et al., 1997). This classifier performs feature selection before generating rules. It is an efficient technique that tries automatically to produce a minimal and a significant set of decision rules without many iterations. Several applications have revealed the need to extend the traditional rough set approach to many variants to handle incomplete and missing databases (Grzymala-Busse, 2003, 2004; Hong et al., 2002; Kryszkiewicz, 1995). This chapter presents the basic concepts of rough sets. Next, we present the two main procedures to create the rough set classifier: the construction and the classification procedures. Finally, some adaptations of rough set classifier are given under imperfect data.

2.2 Basic concepts

2.2.1 Decision table

Data are represented as a table, where each row represents a case, an event, a patient or simply an object. Every column represents an attribute (a variable, an observation, a property, etc.) that can be measured for each object; it can also be supplied by a human expert. Such table is called an information system (information table). It is considered as the basic vehicles for data representation in inductive learning algorithms.

One can define an information table (Pawlak, 1981) in terms of a pair $A = (U, C)$, where $U = \{o_1, o_2, ...o_n\}$ is a non-empty, finite set of objects (cases) called *the universe* and $C = \{c_1, c_2, ...c_k\}$ is a non-empty, finite set of *condition* attributes.

In supervised learning, a special case of information tables is considered, called decision tables (decision systems). A *Decision Table* (DT) is an information system of the form $A =$ (U, C $\cup$ {d}), where $d \notin C$ is a distinguished attribute called *decision*. The value set of d, called $\Theta = \{d_1, d_2, ...d_s\}$.

Example 2.1 *Let us consider a very simple information table shown in Table 2.1. There are six patients (objects) described by the means of three symptoms (attributes): Headache, Muscle-pain and Temperature.*

Table 2.1: Information table

U	Headache	Muscle-pain	Temperature
o_1	no	yes	high
o_2	yes	no	high
o_3	yes	yes	very high
o_4	no	yes	normal
o_5	yes	no	high
o_6	no	yes	very high

Example 2.2 *A small example of DT can be found in Table 2.2. The table includes the same six objects as in Table 2.1 and one decision attribute (Flu) has been added with two possible outcomes: yes, no.*

Table 2.2: Decision table

U	Headache	Muscle-pain	Temperature	Flu
o_1	no	yes	high	yes
o_2	yes	no	high	yes
o_3	yes	yes	very high	yes
o_4	no	yes	normal	no
o_5	yes	no	high	no
o_6	no	yes	very high	yes

The careful reader may notice that objects o_2 and o_5 have exactly the same values of conditions, but they have a different outcome (different value of the decision attribute).

The definitions to be synthesized from decision tables will be of the rule form 'If Headache=no and Muscle-pain=yes and Temperature=high then Flu=yes'. Among the possible properties of the constructed rule sets, minimality is one of the important issues. This is studied in the next subsections.

2.2.2 Indiscernibility relation

A *Decision Table* (DT) expresses all the knowledge about the model. This table may be unnecessarily large. The same or indiscernible objects may be represented several times. We shall look into this issue now. The rough sets adopt the concept of indiscernibility relation (Pawlak, 1982, 1991) to partition the object set U into disjoint subsets, denoted by IND_B or U/B, and the equivalence class that includes o_j is denoted $[o_j]_B$. The objects o_i and o_j are indiscernible on a subset of attributes $B \subseteq C$, if they have the same values for each attribute in subset B of C.

For every object $o_j \in U$, we will use $c_i(o_j)$ to denote the value of a condition attribute c_i for an object o_j. Similarly, $d(o_j)$ is the value of the decision attribute for an object o_j. We further extend these notations for a set of attributes $B \subseteq C$, by defining $B(o_j)$ to be value tuple of attributes in B for an object o_j.

The indiscernibility relation based on a subset of the condition attributes B, denoted by IND_B, is defined as follows:

$$IND_B = U/B = \{[o_j]_B | o_j \in U)\} \tag{2.1}$$

$$where \ \ [o_j]_B = \{o_i | B(o_i) = B(o_j)\} \tag{2.2}$$

The indiscernibility relation based on the decision attribute d, denoted by $IND_{\{d\}}$, is defined as follows:

$$IND_{\{d\}} = U/\{d\} = \{[o_j]_{\{d\}} | o_j \in U)\} \tag{2.3}$$

Example 2.3 *In order to illustrate how a decision table from Table 2.2 defines an indiscernibility relation, we consider the following three non-empty subsets of the conditional attributes: {Headache}, {Headache, Muscle-pain} and {Headache, Muscle-pain, Temperature}. The relation IND defines three partitions of the universe.*

$IND_{\{Headache\}} = \{\{o_2,\, o_3,\, o_5\},\, \{o_1, o_4, o_6\}\}$
$IND_{\{Headache,\, Muscle-pain\}} = \{\{o_1,\, o_4,\, o_6,\},\, \{o_2,\, o_5\},\, \{o_3\}\}$
$IND_{\{Headache,\, Muscle-pain,\, Temperature\}} = \{\{o_1\},\, \{o_2, o_5\},\, \{o_3\},\, \{o_4\},\, \{o_6\}\}$

If we take into consideration the set {Headache}, the objects o_2, o_3 and o_5 belong to the same equivalence class; they are indiscernible.

2.2.3 Set approximation

An equivalence relation induces a partitioning of the universe. It is a natural dimension of reducing data. Since only one element of the equivalence class is needed to represent the entire class. Subsets that are most often of interest have the same value of the outcome attribute. It may happen that a concept such as 'Flu' cannot be defined in a crisp manner. The problematic objects are o_2 and o_5. In other words, it is not possible to induce a crisp description of such objects from table. It is here that the notion of rough sets emerges.

It is possible to delineate the objects that certainly have a positive outcome, the objects that certainly do not have a positive outcome and finally the objects that belong to a boundary between the certain cases. If this boundary is non-empty, the set is rough. These notions are formally expressed as follows (Pawlak, 1982, 1991):

Let $B \subseteq C$ and $X \subseteq U$. We can approximate X using only the information contained by constructing the $B-lower$ and $B-upper$ *approximations* of X, denoted $\underline{\mathrm{B}}$(X) and $\bar{\mathrm{B}}$(X) respectively where

$$\underline{\mathrm{B}}(X) = \{o_j | [o_j]_B \subseteq X\} \tag{2.4}$$

$$\bar{\mathrm{B}}(X) = \{o_j | [o_j]_B \cap X \neq \emptyset\} \tag{2.5}$$

Objects in $\underline{\mathrm{B}}$(X) can be with certainty classified as members of X on the basis of knowledge in B, while objects in $\bar{\mathrm{B}}$(X) can be only classified as possible members of X on the basis of knowledge in B.

The set $BN_B(X)$ is called the $B-$ *boundary region* of X, and thus consists of those objects that we cannot decisively classify into X on the basis of knowledge in B:

$$BN_B(X) = \bar{\mathrm{B}}(\mathrm{X}) - \underline{\mathrm{B}}(\mathrm{X}) \tag{2.6}$$

The set $B-$ *outside region* of X consists of those objects which can be with certainly classified as do not belonging to X, is equal to:

$$U - \bar{\mathrm{B}}(\mathrm{X}) \tag{2.7}$$

A set is said to be rough (respectively crisp) if the boundary region is non-empty (respectively empty).

Example 2.4 *We will continue with the decision table in Table 2.2. Let* $X = \{o_j | Flu(o_j) = yes\}$. *In fact, the set X consists of four objects:* o_1, o_2, o_3 *and* o_6. *Now, we want to describe this set in terms of conditional attributes* $C = \{Headache, Muscle-pain, Temperature\}$. *Using the above definitions, we obtain the following approximation regions: the* $C-$ *lower approximation* $\underline{C}(X) = \{o_1, o_3, o_6\}$, *the* $C-$ *upper approximation* $\bar{C}(X) = \{o_1, o_2, o_3, o_5, o_6\}$, *the* $C-$ *boundary region* $BN_C(X) = \{o_2, o_5\}$ *and the* $C-$ *outside region* $U - \bar{C}(X) = \{o_4\}$. *It is easy to see that the set X is rough since the boundary region is not empty.*

2.2.4 Positive region and dependency of attributes

Another important issue in data analysis is discovering dependencies between attributes. Intuitively, the decision attribute d depends totally on a set of condition attributes C, denoted $C \Rightarrow \{d\}$, if all values of attribute d are uniquely determined by values of attributes from C.

Formally, a functional dependency can be defined in the following way (Pawlak, 1982, 1991). The attribute d depends on C in a degree $k\,(0 \leq k \leq 1)$, denoted $C \Rightarrow_k \{d\}$, if

$$k = \gamma_C(A,\, \{d\}) = \frac{|Pos_C(A,\, \{d\})|}{|U|} \tag{2.8}$$

where

$$Pos_C(A,\, \{d\}) = \bigcup_{X \in U/\{d\}} \underset{\sim}{C}(X), \tag{2.9}$$

$Pos_C(A,\, \{d\})$ is called a positive region of the partition $U/\{d\}$ with respect to C, is the set of all elements of U that can be uniquely classified to blocks of the partition $U/\{d\}$, by means of C.

If $k = 1$ we say that the attribute d depends totally on C, and if k $\prec$ 1, we say that the attribute d depends partially (in a degree k) on the set of attributes C.

The coefficient k expresses the ratio of all elements of the universe, which can be definable classified to blocks of partition $U/\{d\}$, employing attributes C and will be called the *degree of dependency*.

It can be easily seen that if d depends totally on C then $IND_C \subseteq IND_{\{d\}}$. This means that the partition generated by C is finer than the partition generated by d.

The decision attribute d is totally (partially) dependent on the set of the condition attributes C, if employing C all (possible some) elements of the universe U may be uniquely classified to blocks of the partition $U/\{d\}$.

Example 2.5 *Let us consider again a decision table shown in Table 2.2. For example, for dependency {Headache, Muscle-pain, Temperature} $\Rightarrow$ {Flu} we get k=4/6=2/3, because only four patients can be uniquely classified as having flu or not, employing attributes Headache, Muscle-pain and Temperature,* $Pos_C(A,\, \{d\}) = \{o_1,\, o_3,\, o_4,\, o_6\}$.

If we were interested in how exactly patients can be diagnosed using only the attribute Temperature, that is in the degree of dependence {Temperature}

$\Rightarrow \{Flu\}$, we would get k=3/6=1/2, since in this case only patients o_3, o_4 and o_6 out of six can be uniquely classified as having flu. In contrast to the previous case patient o_4 cannot be classified now as having flu or not. Hence the single attribute Temperature offers worse classification than the whole set of attributes Headache, Muscle-pain and Temperature. It is interesting to observe that neither Headache nor Muscle-pain can be used to recognize flu, because for both dependencies $\{Headache\} \Rightarrow \{Flu\}$ and $\{Muscle\text{-}pain\} \Rightarrow \{Flu\}$ we have k=0.

2.2.5 Decision rules

The decision rule induced from a *Decision Table* (DT) is shown as below:

$$\alpha \longrightarrow \beta \text{ with } S$$

-α denotes the conjunction of the conditions that a concept must satisfy.
-β denotes a concept that the rule describes.
-S is a *measure of strength* of which the rule holds.

The support S gives a measure of how trustworthy the rule in drawing conclusion β on the basics of evidence α. Several numerical factors can be associated with a synthesized rule to measure its strength. It can be a frequency based estimate of conditional Probability $Pr(\beta/\alpha)$.

$$S = \frac{|\alpha \cap \beta|}{|\alpha|} \tag{2.10}$$

After the *lower* and the *upper approximations* have been found, the rough set theory can be then used to derive certain and possible rules from them. Rules induced from the lower approximation of the concept certainly describe it, so they are called certain. On the other hand, rules induced from the upper approximation of the concept it only possibly, so they are called possible. The support measure S of a certain rule is equal to 1. The high support measure S for a possible rule the more reliable rule is.

In other words, a decision rule relative to an object o_j is consistent (true, certain), if for every $i \neq j$ $C(o_j) = C(o_i)$ implies $d(o_j) = d(o_i)$. A decision table is consistent if all decision rules are consistent. A decision table is consistent if and only if $C \Longrightarrow \{d\}$. So, there is a relationship between consistency and dependency of attributes in a decision table. If the degree of dependency equals to 1, we conclude that the decision table is consistent.

Example 2.6 *Let us take this rule from the decision table in Table 2.2:*

'If Headache=yes and Muscle-pain=no and Temperature=high then Flu=yes'

The set of objects describes the concept β *is* $\{o_1, o_2, o_3, o_6\}$. *The set of objects describes the left part of the rule* α *is* $\{o_2, o_5\}$. *The strength of the rule is equal to:*

$$S= \frac{|\{o_2, o_5\} \cap \{o_1, o_2, o_3, o_6\}|}{|\{o_2, o_5\}|} = \frac{1}{2}$$

This rule is not true (not certain). So, the decision table is not consistent

2.2.6 Reduct and core

In a previous subsection, we have investigated one natural dimension of reducing data which is to identify equivalence classes, objects that are indiscernible using the available attributes. The other dimension in reduction is to keep only those attributes that preserve the indiscernibility relation and consequently set approximation. The remaining attributes are redundant since their removal does not worsen the classification. There is usually several such subsets of attributes and those which are minimal are called reducts. In order to express the above idea more precisely we need some auxiliary notions:

Let $c \in C$, the attribute c is dispensable in C with respect to d, if $Pos_C(A, \{d\}) = Pos_{C-c}(A, \{d\})$. Otherwise attribute c is indispensable in C with respect to d.

If all attribute $c \in C$ are indispensable in C with respect to d, then C will be called independent.

Reduct

A subset $B \subseteq C$ is a reduct of C with respect to d, if B is independent and:

$$Pos_B(A, \{d\}) = Pos_C(A, \{d\}) \tag{2.11}$$

Hence, a reduct is a set of attributes from C that preserves partition and, consequently, set approximation. It means that a reduct is the minimal subset of attributes that enables the same classification of elements of the universe as the whole set of attributes. In other words, attributes that do not belong to a reduct are superfluous with regard to classification of elements of the universe.

Note that computing equivalence classes is straightforward. Finding a minimal reduct (reduct with a minimal number of attributes) among all reduct is NP-hard. This means that computing reducts is not a trivial task that cannot be solved by a simple increase of computational resources. It is, in fact, one of the drawbacks of rough set methodology. Fortunately, there exist good heuristics (Chouchoulas & Shen, 2001; Jensen & Shen, 2003; Wroblewski, 1995; Zhong et al., 2001) that compute sufficiently many reducts in often acceptable time, unless the number of attributes is very high.

Core

The set of all the condition attributes indispensable in C with respect to d is denoted by $Core_C(A, \{d\})$. It is the intersection of all reducts of C.

$$Core_C(A, \{d\}) = \bigcap Red_C(A, \{d\}) \tag{2.12}$$

Where $Red_C(A, \{d\})$ is the set of all reducts of C.

Since the core is the intersection of all reducts, it is included in every reduct. Thus, in a sense, the core is the most important subset of attributes, for none of its elements can be removed without affecting the classification power of attributes.

We will need also a concept of a value reduct and value core. Suppose we are given a dependency $C \Rightarrow \{d\}$. To further investigation of the dependency, we might be interested to know exactly how values of attribute d depend on values of attributes from C. To this end, we need a procedure of eliminating values of attributes form C which do not influence on values of attribute d. This issue will be detailed in the next subsection.

Example 2.7 *Let us continue with the decision system in Table 2.2, to compute the possible reducts and core.*

$Pos_C(A, \{d\})= \{o_1, o_3, o_4, o_6\}$
$Pos_{\{Headache\}}(A, \{d\})=\emptyset$
$Pos_{\{Muscle-pain\}}(A, \{d\})=\emptyset$

$Pos_{\{Temperature\}}(A, \{d\})=\{o_3, o_4, o_6\}$
$Pos_{\{Headache,Muscle-pain\}}(A, \{d\})=\{o_3\}$
$Pos_{\{Headache,Temperature\}}(A, \{d\})=\{o_1, o_3, o_4, o_6\}$
$Pos_{\{Muscle-pain,Temperature\}}(A, \{d\})=\{o_1, o_3, o_4, o_6\}$

In Table 2.2, there are two possible reducts with respect to Flu, {Muscle-pain, Temperature} and {Headache, Temperature} are independent with respect to the decision Flu and have the same positive region that the whole subset of condition attributes C. That means that either the attribute $Muscle-pain$ or $Headache$ can be eliminated from the table and consequently instead of Table 2.2, we can use either Table 2.3 or Table 2.4. The core is the attribute $Temperature$. It is the intersection of the two possible reducts.

Table 2.3: First reduct

U	Headache	Temperature	Flu
o_1	no	high	yes
o_2	yes	high	yes
o_3	yes	very high	yes
o_4	no	normal	no
o_5	yes	high	no
o_6	no	very high	yes

Table 2.4: Second reduct

U	Muscle-pain	Temperature	Flu
o_1	yes	high	yes
o_2	no	high	yes
o_3	yes	very high	yes
o_4	yes	normal	no
o_5	no	high	no
o_6	yes	very high	yes

For Table 2.2, the core with respect to the set {Headache, Muscle-pain, Temperature} is the Temperature. This confirms that Temperature is the only symptom that enables, at least, partial diagnosis of patients.

2.2.7 Value reduct and value core

To further simplify the decision table, we can eliminate some values of attribute from the table. To this end, we can apply similar procedure as eliminate superfluous attributes, which is defined next.

We will say that the value of attribute $c \in C$, is dispensable for o_j with respect to d, if $[o_j]_C \subseteq [o_j]_{\{d\}}$ implies $[o_j]_{C-\{c\}} \subseteq [o_j]_{\{d\}}$; otherwise the value of attribute c is indispensable for o_j with respect to d.

If for every attribute $c \in C$ the value of c is indispensable for o_j, then C will be called independent (orthogonal) for o_j.

A subset $B \subseteq C$ is a value reduct of C for o_j, if B is orthogonal for o_j and $[o_j]_C \subseteq [o_j]_{\{d\}}$ implies $[o_j]_B \subseteq [o_j]_{\{d\}}$.

The set of all indispensable values of attributes from C for o_j will be called core of C for o_j (the value core), and will be denoted $Core_C^j(A, \{d\})$.

We have also the following property

$$Core_C^j(A, \{d\}) = \bigcap Red_C^j(A, \{d\}) \tag{2.13}$$

Where $Red_C^j(A, \{d\})$ is the family of all reduct of C for o_j with respect to d.

Example 2.8 *Let us continue with the decision system in Table 2.2. Using the concept of a value reduct. Table 2.3 and Table 2.4 can be simplified as follows:*

Table 2.5: Reduced decision table 1

U	Headache	Temperature	Flu
o_1	no	high	yes
o_2	yes	high	yes
o_3	-	very high	yes
o_4	-	normal	no
o_5	yes	high	no
o_6	-	very high	yes

Table 2.6: Reduced decision table 2

U	Muscle-pain	Temperature	Flu
o_1	yes	high	yes
o_2	no	high	yes
o_3	-	very high	yes
o_4	-	normal	no
o_5	no	high	no
o_6	-	very high	yes

In order to explain how we can obtain Tables 2.5 and 2.6, we only take the third rule relative to the object o_3 from Table 2.3 and we compute their reduct values. The possible reduct values are $Headache = yes$ and $Temperature = very\,high$.

We check if: $[o_3]_{\{Headache\}} \subseteq [o_3]_{\{d\}}$
$\{o_2, o_3, o_5\} \nsubseteq \{o_1, o_2, o_3, o_6\}$
We check if: $[o_3]_{\{Temperature\}} \subseteq [o_3]_{\{d\}}$
$\{o_3,\, o_6\} \subseteq \{o_1, o_2, o_3, o_6\}$

So, $Temperature = very\,high$ is the only value reduct for the object o_3 with respect to Flu and it is also the value core.

Discernibility matrix and function

In order to easily compute reducts and core or value reduct and value core, we can use the discernibility matrix (Skowron & Rauszer, 1992) which is defined below. Let A be a decision table with n objects. The discernibility matrix of A is a symmetric n*n matrix with entries a_{ij} as given below. Each entry thus consists of the set of attributes upon which objects o_i and o_j differ.

$$a_{ij} = \{c \in C \backslash c(o_i) \neq c(o_j)\}\ and\ d(o_i) \neq d(o_j)\ for\, i, j = 1, ..., n \qquad (2.14)$$

Thus, entry a_{ij} is the set of all attributes which discern objects o_i and o_j that do not belong to the same equivalence class $IND_{\{d\}}$.

A discernibility function f_A for a decision table A is a boolean function of m boolean variables $c_1^*...c_m^*$ (corresponding to the attributes $c_1...c_m$) defined as below, where $a_{ij}^* = \{c^*|c \in a_{ij}\}$

$$f_A(c_1^*, ..., c_m^*) = \wedge\{\vee a_{ij}^*|1 \leq j \leq i \leq n, a_{ij} \neq \emptyset\} \tag{2.15}$$

where $\wedge$ and $\vee$ are two logical operators for conjunction and disjunction. The set of all prime implicants[1] of f_A determines the sets of all reducts of A.

If we construct a boolean function by restricting the conjunction to only run over column j in the discernibility matrix (instead of over all columns), we obtain the so-called j-relative discernibility function. The set of all prime implicants of this function determines the set of all j-reducts of A (the value reduct for the object o_j). The value reducts reveal the minimum amount of information needed to discern $o_j \in U$.

Example 2.9 *Let us continue with the same decision table in Table 2.2. We will use the notations H, M and T respectively for $Headache$, $Muscle-pain$ and $Temperature$. The discernibility matrix and function are:*

Table 2.7: Discernibility matrix

	o_1	o_2	o_3	o_4	o_5	o_6
o_1						
o_2						
o_3						
o_4	T	H,M,T				
o_5	H,M		M,T			
o_6				T	H,M,T	

$$f_A(H, M, T) = T \wedge (H \vee M) \wedge (H \vee M \vee T) \wedge (M \vee T)$$

[1] An implicant of a Boolean function f is any conjunction of literals (variables or their negations) such that if the values of these literals are true under an arbitrary valuation v of variables then the value of the function f under v is also true. A prime implicant is a minimal implicant. Here, we are interested in implicants of monotone Boolean functions only (functions constructed without negation).

where each parenthesized tuple is a conjunction in the Boolean expression, and where the one-letter Boolean variables correspond to the attribute names in an obvious way. After the simplification of the discernibility function, we obtain the following expression:

$$f_A(H, M, T) = (T \wedge H) \vee (T \wedge M)$$

Which represents two reducts {$Temperature$, $Headache$} and {Temperature, Muscle-pain} in the decision table where $Temperature$ is the core.

2.3 Rough Set Classifier (RSC)

A *Rough Set Classifier* (RSC) is an interesting classification technique applied in many fields like medicine, finance, telecommunication, intelligent agents, image analysis, pattern recognition, process industry, marketing, etc. The models generated by the rough set classifier take the form of 'IF-THEN' rules, which have the advantage of interpretation. Besides, feature selection is a very critical step when building the model which decreases the time complexity of learning, size of induced decision rules and improves their qualities. Several software systems have been developed that support the process of constructing and validating rough set classifier providing advanced graphical environments, for instance, KDD-R (Ziarko, 1998), LERS (Grzymala-Busse, 1992), ProbRough (Piasta & Lenarcik, 1996), etc.

A *Rough Set Classifier* (RSC) is made using two different procedures:

1. Construction procedure
2. Classification procedure

2.3.1 RSC: Construction procedure

The construction procedure of the RSC is described by the mean of the following phases:

1. Creation of the training decision table.
2. Discretization of real values attributes.
3. Simplification of the training decision table.
4. Generation of the decision rules.

We will detail these phases in the following parts:

Phase 1. Creation of the training decision table

The raw input data set is transformed into a decision table where rows correspond to objects, and columns correspond to features. The decision table is subsequently split into two parts: the training dataset (a set of instances used for learning the model generated by the classifier) and the testing dataset (a set of instances used to evaluate the performance of the classifier). So, the rough set classifier will be induced from the training dataset and applied to the testing dataset to obtain a performance estimate.

Example 2.10 *Let us take an example of training decision system shown in Table 2.8 where a, b, c, d are four condition attributes and e is a decision attribute with three possible values* $\Theta = \{0, 1, 2\}$. *The condition attributes b,c,d are symbolic. However, the attribute a is real valued. We need in this case a discretization process. We look to this issue in the next phase.*

Table 2.8: Training decision table

U	a	b	c	d	e
o_1	1.3	0	0	1	1
o_2	1.3	0	0	0	1
o_3	0.8	0	0	0	0
o_4	1.4	1	0	1	0
o_5	1.4	1	0	2	2
o_6	1.6	1	0	2	2
o_7	1.6	2	2	2	2
o_8	1.3	0	0	0	1

Phase 2. Discretization of real values attributes

A real-world data set always contains mixed types of data such as continuous valued, symbolic data, etc. When it comes to analyze attributes with real values, they must undergo a process called discretization, which divides the attributes value into intervals. There is a lack of the unified approach to discretization problems so far, and the choice of method depends heavily on data considered. Discretization is a step that is not specific to the rough set approach but that most rule induction algorithms currently require for them to perform well. A number of successful approaches to the problem of finding effective methods for real value attributes quantization have been proposed by machine learning. The rough set community has been also committed to constructing efficient algorithms for discretization such as a discretization system called the RSBR that is based on hybridization of *Rough Sets* and *Boolean Reasoning* proposed in (Nguyen, 1997).

Example 2.11 *Let us continue with the Table 2.8 to discretize the attribute a based on RSBR. The main steps of the algorithm can be described as follows:*

Step 1. Define a set of Boolean Variables BV(U). We have $BV(U) = \{p_1^a, p_2^a, p_3^a, p_4^a\}$, *where* p_1^a *corresponds to the interval [0.8, 1) of a;* p_2^a *corresponds to the interval [1,1.3) of a;* p_3^a *corresponds to the interval [1.3, 1.4) of a;* p_4^a *corresponds to the interval [1.4, 1.6) of a.*

Step 2. Create a new decision table A^p *by using the set of Boolean variables defined in Step 1. Here* A^p *is called the P-discretization of A,* $A^p = (U, \cup\{d\}, p_k^c)$, p_k^c *is a propositional variable corresponding to the interval* $[v_k^c, p_{k+1}^c)$ *for any* $k \in \{1,\ldots,n_c\text{-}1\}$ *and* $c \in C$.

Table 2.9 shows an example of A^p. *We set,* $p_1^a(o_1, o_2) = 1$, *because any cut in the interval [0.8, 1) corresponding to* p_a^1 *discerns* o_1 *and* o_2.

Table 2.9: An example of A^p

U	p_1^a	p_2^a	p_3^a	p_4^a
(o_1, o_2)	1	0	0	0
(o_1, o_3)	1	1	0	0
(o_1, o_5)	1	1	1	0
(o_4, o_2)	0	1	1	0
(o_4, o_3)	0	0	1	0
(o_4, o_5)	0	0	0	0
(o_6, o_2)	0	1	1	1
(o_6, o_3)	0	0	1	1
(o_6, o_5)	0	0	0	1
(o_7, o_2)	0	1	0	0
(o_7, o_3)	0	0	0	0
(o_7, o_5)	0	0	1	0

Step 3. Find a minimal subset of P that discerns all the objects in different decision classes by using the discernibility formula.

$$\Phi^U = \wedge\{\psi(i,j) : d(o_i) \neq d(o_j)\} \tag{2.16}$$

Where, $\psi(i,j) = p_1^a$ *means that in order to discern object* o_1 *and* o_2*, the cut between a(0.8) and a(1) must be selected. From Table 2.9, we obtain the discernibility formula:*

$\Phi^U = (p_1^a) \wedge (p_1^a \vee p_2^a) \wedge (p_1^a \vee p_2^a \vee p_3^a) \wedge (p_2^a \vee p_3^a) \wedge (p_2^a) \wedge (p_2^a \vee p_3^a \vee p_4^a) \wedge (p_3^a \vee p_4^a) \wedge (p_4^a) \wedge (p_2^a) \wedge (p_3^a)$

Finally, we obtain four prime implicants of the discernibility formula,

$\Phi^U = (p_1^a \vee p_4^a) \wedge (p_2^a) \wedge (p_3^a) \wedge (p_3^a)$

Furthermore, we select $\{p_1^a, p_4^a\}$*, i.e.* P = *{(a, 0.9); (a, 1.5)} as the optimal result, because it is the minimal subset of* P *preserving discernibility.*

Table 2.10: P-discretization of the training decision table

U	a	b	c	d	e
o_1	1	0	0	1	1
o_2	1	0	0	0	1
o_3	0	0	0	0	0
o_4	1	1	0	1	0
o_5	1	1	0	2	2
o_6	2	1	0	2	2
o_7	2	2	2	2	2
o_8	1	0	0	0	1

Phase 3. Simplification of the training decision table

Real-world databases are characterized by a lot of redundant and irrelevant informations having no effect on classification performance. If these redundancies are not removed, not only the time complexity of rule discovery increases, but also the quality of the discovered rules may be significantly depleted. Hence, the simplification of decision table has a primary importance in many applications. It has been investigated by many researchers and there are a variety of informal approaches to this problem. The approach to table simplification presented here consists of the following steps (Pawlak, 1982, 1991):

Step. 1 Elimination of the superfluous condition attributes: An example of simplification is the reduction of condition attributes in a decision table called attribute selection. In the reduced decision table, the same decisions can be based on a smaller number of conditions. This kind of simplification eliminates the need for checking unnecessary conditions or, in some applications, for performing, expensive tests to arrive at a conclusion which eventually could be achieved by simpler means.

There are several attempts to solve this problem based on rough sets (Modrzejewski, 1993; Pawlak, 1991; Rauszer, 1991). One of the ideas was to consider as relevant features those in reducts of the decision table. Computation of reducts is equivalent to eliminate some of condition attributes (columns) from the decision table.

Example 2.12 *Let us continue with the same example in Table 2.8. It is easy to compute that only dispensable attribute is c with respect to e; consequently, we can remove column c in Table 2.8, which yields Table 2.11 shown below.*

Table 2.11: Reduct of the training decision table

U	a	b	d	e
o_1	1	0	1	1
o_2	1	0	0	1
o_3	0	0	0	0
o_4	1	1	1	0
o_5	1	1	2	2
o_6	2	1	2	2
o_7	2	2	2	2
o_8	1	0	0	1

Step 2. Elimination of the redundant objects: After selecting the more relevant features from conditional attributes, we will find duplicate objects (rows) in the decision table. Duplicate rows have the same value of

condition and decision attributes.

Example 2.13 *In Table 2.11, there are two redundant objects o_2 and o_8. Hence, we keep the object o_2 and we delete the object o_8.*

Step 3. Elimination of the superfluous values of condition attributes: A reduced table can be seen as a rule set where each rule corresponds to one object of the table. The rule set can be generalized further by applying rough set value reduction method. The main idea behind this method is to drop those redundant condition values of rules and to unite those rules in the same decision. Hence, after removing the superfluous attributes and redundant objects, we can remove some superfluous attributes values for some rule using the concept of value reduct and value core.

Example 2.14 *In this step we have to reduce superfluous values of condition attributes, in every decision rule. We have first to compute core values of condition attributes in every decision rule. For the sake of illustration, let us compute the core values of condition attributes for the first object o_1:*

'If a=1 and b=0 and d=1 then e=1'.

If we remove the value a=1 from the rule there are no inconsistency. So, a=1 is dispensable with respect to the decision e. It is not a core value.
If we remove the value d=1 from the rule there are no inconsistency. So, d=1 is dispensable with respect to the decision e. It is not a core value.
If we remove the value b=0 from the rule there are inconsistencies :

1. *'If a=1 and d=1 then e=1',*
2. *'If a=1 and d=1 then e=0'.*

So, b=0 is dispensable with respect to the decision e. It is the only core value.

Similarly we can compute remaining core values of condition attributes in every decision rule and the final results are presented in Table 2.12.

Having computed core values of condition attributes, we can proceed to compute value reducts. As an example, let us compute value reduct for the object o_1 of the decision table. The set of possible reducts are:

'If $b = 0$ then $e = 1$'. It is not consistent.

Table 2.12: Core values of the decision rules

U	a	b	d	e
o_1	-	0	-	1
o_2	1	-	-	1
o_3	0	-	-	0
o_4	-	1	1	0
o_5	-	-	2	2
o_6	-	-	-	2
o_7	-	-	-	2

'If $b = 0$ and $d = 1$ then $e = 1$'. It is consistent.
'If $a = 1$ and $b = 0$ then $e = 1$'. It is consistent.

Hence, we have two value reducts $b = 0$ and $d = 1$ or $a = 1$ and $b = 0$.

In Table 2.13 below we list value reducts for all decision rules.

Table 2.13: Reduct values of the decision rules

U	a	b	d	e
o_1	1	0	-	1
o_1'	-	0	1	1
o_2	1	0	-	1
o_2'	1	-	0	1
o_3	0	-	-	0
o_4	-	1	1	0
o_5	-	-	2	2
o_6	-	-	2	2
o_6	2	-	-	2
o_7	-	-	2	2
o_7'	-	2	-	2
o_7"	2	-	-	2

Phase 4. Generation of the decision rules

The reader has certainty realized that the reducts and reduct values can be used to synthesize minimal decision rules. Once they have been computed, the rules are easily constructed by reading off the values. The rule derived from this stage can be used to classify the data. The set of rules is referred to as a classifier and can be used to classify new and unseen data.

Example 2.15 *As we can see from Table 2.13, for decision rules 1 and 2, we have two value reducts of condition attributes. Decision rules 3, 4 and 5 have only one value reducts of condition attributes for each decision rule row. The remaining decision rules 6 and 7 contain two and three value reducts respectively. Thus, there are 2*2*2*3=24 (not necessarily) solutions to our problem. Since decision rules 1 and 2 are identical, and so are rules 5, 6 and 7, we can represent our table in the form :*

Table 2.14: Simplified decision rules

U	a	b	d	e
o_1, o_2	1	0	-	1
o_3	0	-	-	0
o_4	-	1	1	0
o_5, o_6, o_7	-	-	2	2

This solution will be referred to as minimal. Looking in Table 2.14, the induced decision rules are:

'If a=1 and b=0 then e=1',
'If a=0 then e=0',
'If b=1 and d=1 then e=0',
'If d=2 then e=2'.

All the induced decision rules from Table 2.14 are true.

2.3.2 RSC: Classification procedure

To classify objects, which have never been seen before, decision rules generated from the training set will be used. These rules represent the actual classifier. This classifier is used to predict to which class (decision) new object is attached. The same cuts computed from training dataset discretization method are first used to discretize the new object dataset.

Let us consider one of the simplest application scheme which has shown to be useful in practice.

- When a rough set classifier is presented with a new case, the rule set is scanned to find applicable rules. Rules whose predecessors match the case.
- If no rule is found, the most frequent outcome in the training data is chosen.
- If more than one rule fires, these may in turn indicate more than possible outcome. A voting process is then performed among the rules that fire in order to solve conflicts and to rank the predicted outcomes. The new object will be assigned to the decision class with maximal strength of the selected rule set.

For a systematic overview of rule synthesis see (Grzymala-Busse & Wang, 1997; Skowron, 1995; Stefanowski, 1998).

2.4 Rough set classifier for incomplete data

Data sets can be roughly classified into two categories: complete and incomplete data sets. All objects in complete data sets have known attribute values. If at least one object in data set has a missing or imprecise value, the set is incomplete or uncertain.

Learning from incomplete or uncertain data sets is usually more difficult than from complete data sets. Several extensions of induction systems based on rough sets have been developed to handle the problem of incomplete data sets with missing attribute values (Grzymala-Busse, 2003, 2004; Hong et al., 2002; Kryszkiewicz, 1995, 1999; Lingras, 1995; Stefanowski, 2001; Stefanowski & Tsoukias, 2001).

All these extensions deal with incomplete and missing data, but not with partially uncertain data. There are two kinds of missing data existing in decision table:

1. **Lost value:** where the original value was erased, represented by the symbol '?'.

2. **Do not care value:** where the original value was irrelevant, represented by the symbol '*'.

Incomplete decision tables in which all attribute values are lost, from the viewpoint of the rough set theory, were studied for the first time in (Grzymala-Busse & Wang, 1997) where two algorithms for rule induction, modified to handle lost attribute values, were presented. This approach was studied later in (Stefanowski, 2001; Stefanowski & Tsoukias, 2001), where the indiscernibility relation was generalized to describe such incomplete decision tables.

Incomplete decision tables in which all missing attribute values are 'Do not care' conditions, from the view point of rough set theory, were attempted for the first time in (Grzymala-Busse, 1991), where a method for rule induction was introduced in which missing attribute values were replaced by all values from the domain of the attribute. 'Do not care' conditions were also studied later in (Kryszkiewicz, 1995, 1999), where the indiscernibility relation was again generalized, this time to describe incomplete decision tables with 'do not care' conditions.

In general, incomplete decision tables are described by characteristic relations, in a similar way as complete decision tables are described by indiscernibility relations (Grzymala-Busse, 2003, 2004). In rough set theory, one of the basic notions is the idea of lower and upper approximations. For complete decision tables, once the indiscernibility relation is fixed and the concept is given, the lower and upper approximations are unique. For incomplete decision tables, for a given characteristic relation and a concept, there are three different possibilities to define the lower and the upper approximations, called singleton, subset and concept approximations. Only the latter is applicable in data mining (Grzymala-Busse, 2003, 2004).

Some other works have combined the rough set theory with other theories of uncertainty to handle the problem of incomplete data such as probability theory, fuzzy set theory and belief function theory.

2.4.1 Probability theory and rough set theory

Piasta and Lenarcik (1996) present an algorithm ProbRough for inducing decision rules from data. The algorithm combines all the positive aspects of rule induction systems with the flexibility of the probabilistic representation of data and learned knowledge. It accepts imprecise and incomplete data of any mixed qualitative and quantitative type. The probabilistic structure of data is estimated from the learning set using the frequency-based estimators. The results of learning are described in the form of a certain partition of the feature space. The partition elements are of a special geometrical shape, which enables the presentation of the decision rules in a simple form meaningful to humans. Nevertheless, the prediction accuracy is comparable or superior to more sophisticated classifiers. The ProbRough algorithm searches through the set of various partitions using the criterion based on minimizing the misclassification costs.

2.4.2 Fuzzy set theory and rough set theory

In (Hong et al., 2002), the problem of producing a set of certain and possible rules from incomplete quantitative data is handled. The rough set theory and the fuzzy set concepts are combined to solve this problem. A new generalized fuzzy learning algorithm based on fuzzy incomplete equivalence classes proposed to simultaneously derive certain and possible fuzzy rules from incomplete quantitative data sets and estimate the missing values in the learning process.

2.4.3 Belief function theory and rough set theory

In (Lingras, 1995), a rough set based methodology for extracting rules from incomplete databases was proposed. Conventionally, the rough set model is used for rule extraction from databases where some values of the attributes are precisely known. The proposed approach is a generalization of existing approach by allowing for a set of possible attribute values. Such an approach can be useful in situations where the value of an attribute may be one of a few possible choices. The developed work suggests a relaxation of some of the assumptions in the rough set theory as well as the resulting plausibility functions. The rule extraction process described in this study is based on belief functions.

2.5 Conclusion

The rough set theory is applied in many real-world applications such as medical data analysis, finance, voice recognition, image processing and others. The standard version of rough set is too simple for many real life applications. This chapter presented the basic concepts of this theory, its role in KDD process and some extensions and variants proposed by various researchers.

All extensions of the rough set classifier handle only incomplete decision table characterized by some missing attribute values and not with partially uncertain decision attribute values. Our work also presented other extensions of rough set classification from partially uncertain data where the uncertainty exists only in decision attribute values and is handled by the TBM. Our new solutions, denoted by *Belief Rough Set Classifier* (BRSC) and *Belief Rough Set Classifier* based on *Generalization Distribution Table* (BRSC-GDT), will be detailed in the next part.

Part II

Belief Rough Set Classifier

In this second part, considered as the major of our thesis, we detail the developments that we have proposed in order to build our two classification approaches called the *Belief Rough Set Classifier* (BRSC) and the *Belief Rough Set Classifier* based on *Generalization Distribution Table* (BRSC-GDT) from *Uncertain Decision Table* (UDT). The uncertainty exists only in the decision attribute and is handled by the TBM, one interpretation of the belief function theory. We only handle symbolic condition and decision attribute values. This part is composed of four chapters:

- Chapter 3 describes the basic concepts of rough sets under the belief function framework. These new definitions of the basic concepts are needed to create our first classification approach called BRSC. We also detail the basic notions of the hybrid system GDT-RS needed to create the second classification approach called BRSC-GDT. The latter was also generalized under the new context to be called belief GDT-RS.

- Chapter 4 presents the two main procedures of the belief rough set classifier and the belief rough set classifier based on generalization distribution table: the construction and the classification procedures. The first procedure consists of the main phases needed to create both the classifiers and the second procedure deals with the classification of new objects.

- Chapter 5 presents some ideas to improve the performance of our classification systems (BRSC, BRSC-GDT). To reduce the time requirement needed to build the two models, we propose a heuristic attribute selection method from the partially uncertain data. This heuristic method selects the more relevant condition attributes without costly calculation. To obtain more stable results from the two classifiers, we propose the concept of dynamic reduct from uncertain data to select the more efficient and stable attributes from noisy and uncertain data.

- Chapter 6 presents the experimental phase which is performed in order to check the feasibility of our approaches and judge their qualities. The time requirement, the size of the model and the accuracy are the evaluation criteria . Results obtained from real-world databases are analyzed and compared with those obtained from a similar classifier called *Belief Decision Tree* (BDT).

Chapter 3

Basic concepts of belief rough set classifiers

3.1 Introduction

The objective of our thesis is to learn decision rules from uncertain data to classify new objects. Hence, we propose in our work two classification approaches based on *Rough Sets* (RS) that are able to generate decision rules from uncertain data. We assume that the uncertainty exists only in the decision attribute values of the *Decision Table* (DT) and is represented by the belief functions. The first classification technique, named *Belief Rough Set Classifier* (BRSC), is only based on the basic concepts of the *Rough Sets* (RS). The second is more sophisticated, called *Belief Rough Set Classifier* based on *Generalization Distribution Table* (BRSC-GDT), and derived from the hybrid system GDT-RS. The latter is a combination of the *Generalization Distribution Table* (GDT) and the *Rough Set* methodology (RS).

In the first part of this chapter, we describe the modified basic concepts of rough sets under belief function framework like uncertain decision table and tolerance relation. These concepts are useful to build the BRSC. The second part of this chapter presents the basic notions of the hybrid system GDT-RS and its new definition under the belief function framework called belief GDT-RS. It is needed to build the BRSC-GDT.

3.2 Basic concepts of rough sets under belief functions

One issue in real-world databases is the uncertainty. This kind of data exists in many real-world applications like in medicine where symptoms or diseases of some patients may be totally or partially uncertain. It is not efficient to eliminate these objects from classification process because it will result in a loss of important information. Many works have been done to adapt rough sets to this kind of environment (Grzymala-Busse, 2003, 2004; Hong et al., 2002; Kryszkiewicz, 1995; Stefanowski, 2001; Stefanowski & Tsoukias, 2001). These extensions deal with incomplete decision tables which may be characterized by missing condition attribute values, but not with uncertain decision attributes. Due to this shortcoming, we propose in this thesis our two classification approaches the BRSC and the BRSC-GDT which are able to learn decision rules from uncertain data in the objective to classify unseen objects. The uncertainty only exists in the decision attribute of the training sets. This kind of uncertainty can be represented by the theory of belief functions. It is considered as a useful theory for representing partial uncertain knowledge. In this work, we use the TBM, one interpretation of belief function theory (Smets & Kennes, 1994; Smets, 1998a). In order to create our new classification systems based on rough sets especially the BRSC, we need to redefine the basic concepts of rough sets in the uncertain context.

This section describes the modified definitions of an uncertain decision table, tolerance relation, set approximation, positive region, dependency of attributes and especially core and reduct needed in the construction procedure of BRSC. These news definitions were originally proposed in (Trabelsi & Elouedi, 2008, 2009).

3.2.1 Uncertain decision table

Our *Uncertain Decision Table* (UDT) is defined as follows: $A = (U, C \cup \{ud\})$, where $U = \{o_1, o_2, ...o_n\}$ is a finite set of n objects characterized by a set of k certain condition attributes $C = \{c_1, c_2, ...c_k\}$ and an uncertain decision attribute $ud \notin C$.

We propose to represent the uncertainty of each object o_j by a bba m_j expressing belief on decisions defined on the frame of discernment $\Theta = \{ud_1, ud_2, ...ud_s\}$ representing the s possible values of the decision attribute ud. These bba's are generally given by an expert (or several experts) to express partially uncertain decision attribute values, they can also present the two extreme cases of total knowledge and total ignorance.

Example 3.1 *Let us take Table 3.1 to describe our uncertain decision table. The latter contains eight objects (patients), three certain condition attributes* C=$\{Headache,\ Muscle-pain,\ Temperature\}$ *and an uncertain decision attribute* $ud = Flu$ *with two possible values* $\{yes,\ no\}$ *representing* Θ.

Table 3.1: Uncertain decision table 1

U	Headache	Muscle-pain	Temperature	Flu
o_1	yes	yes	very high	$m_1(\{yes\}) = 0.95\ \ m_1(\{no\}) = 0.05$
o_2	yes	no	high	$m_2(\{no\}) = 1$
o_3	yes	yes	normal	$m_3(\{yes\}) = 0.5\ \ m_3(\Theta) = 0.5$
o_4	no	yes	normal	$m_4(\{no\}) = 0.6\ \ m_4(\Theta) = 0.4$
o_5	no	yes	normal	$m_5(\{no\}) = 1$
o_6	yes	no	high	$m_6(\{no\}) = 0.95\ \ m_6(\Theta) = 0.05$
o_7	no	yes	very high	$m_7(\{yes\}) = 1$
o_8	no	yes	high	$m_8(\{yes\}) = 0.9\ m_8(\Theta) = 0.1$

For the object o_3*, 0.5 of beliefs are exactly committed to the decision yes, whereas 0.5 of beliefs is assigned to the whole of frame of discernment* Θ *(ignorance). With bba, we can represent the certain case (with a certain decision) like for the objects* o_2*,* o_5 *and* o_7*.*

3.2.2 Tolerance relation

The standard rough sets adopt the concept of indiscernibility relation (Pawlak, 1982, 1991) to partition the object set U into disjoint subsets (equivalence classes), denoted by U/B. The objects o_i and o_j are indiscernible on a subset of attributes B, if they have the same values for each attribute in subset B.

In our uncertain context, the indiscernibility relation for the condition attributes U/C is the same as in the certain case because their values are certain and it is computed using eqn.(2.1).

Example 3.2 *Let us continue with the same uncertain decision system in Table 3.1 to compute the equivalence classes based on condition attributes in the same manner as in the certain case:*

$U/C = \{\{o_1\}, \{o_2, o_6\}, \{o_3\}, \{o_4, o_5\}, \{o_7\}, \{o_8\}\}$.

Contrary to the condition attributes, the indiscernibility relation $U/\{ud\}$ for the uncertain decision attribute ud is not the same as in the certain case. The decision value is represented by a bba. For this reason, the indiscernibility relation will be called *tolerance relation* and denoted by Tol_{ud}. It should be noted here that the term equivalence class from the certain decision attribute case will be replaced by tolerance class for the uncertain decision attribute, because the resulting classes may overlap.

The number of tolerance classes is known. The latter represents the possible values of ud. So, we need for optimal decision making assign each object o_j, characterized by a bba m_j, to the right tolerance class X_i relative to the decision value ud_i.

To solve this issue, we propose two alternatives described in the following parts:

First alternative: restrictive solution

The first idea is to use the pignistic transformation. It is a function which can transform the belief function to probability function in order to make decisions from beliefs using eqn. (1.55). We suggest, for each object o_j in the uncertain decision table, compute the pignistic probability, denoted $BetP_j$, by applying the pignistic transformation to m_j.

For every ud_i, we define a tolerance class X_i as follows:

$$X_i = \{o_j | BetP_j(\{ud_i\}) > 0\} \tag{3.1}$$

Besides, we define a tolerance relation $Tol_{\{ud\}}$ as follows:

$$Tol_{\{ud\}} = U/\{ud\} = \{X_i | ud_i \in \Theta\} \tag{3.2}$$

Example 3.3 *Let us continue with the same example to compute the tolerance classes based on the uncertain decision attribute $U/\{ud\}$ by applying* *the first alternative as follows:*

Table 3.2: Pignistic transformation to m_j for o_j

m_j	$BetP_j$
m_1	$BetP_1(\{yes\}) = 0.95$ $BetP_1(\{no\}) = 0.05$
m_2	$BetP_2(\{yes\}) = 0$ $BetP_2(\{no\}) = 1$
m_3	$BetP_3(\{yes\}) = 0.75$ $BetP_3(\{no\}) = 0.25$
m_4	$BetP_4(\{yes\}) = 0.2$ $BetP_4(\{no\}) = 0.8$
m_5	$BetP_5(\{yes\}) = 0$ $BetP_5(\{no\}) = 1$
m_6	$BetP_6(\{yes\}) = 0.975$ $BetP_6(\{no\}) = 0.025$
m_7	$BetP_7(\{yes\}) = 1$ $BetP_7(\{no\}) = 0$
m_8	$BetP_8(\{yes\}) = 0.95$ $BetP_8(\{no\}) = 0.05$

Table 3.2 shows the pignistic probability applying to each m_j. *According to it, the object* o_7 *is assigned only to the tolerance class* X_1 *relative to the decision value* $ud_1 = yes$. *The objects* o_2 *and* o_5 *are assigned only to the tolerance class* X_2 *relative to the decision value* $ud_2 = no$. *The objects* o_1, o_3, o_4, o_6 *and* o_8 *are included in the two tolerance classes. So, the tolerance relation* Tol_{ud} *based on the uncertain decision attribute ud is equal to:*

$$U/\{ud\}=\{\{o_1, o_3, o_4, o_6, o_7, o_8\}, \{o_1, o_2, o_3, o_4, o_5, o_6, o_8\}\}.$$

We can conclude that the first alternative is too restrictive. The objects o_1, o_6 *and* o_8 *are near from the certain case. However, they exist in the two tolerance classes* X_1 *and* X_2.

Second alternative: flexible solution

Let us remember that we would like to assign each object to the right tolerance class X_i relative to the decision value ud_i according to its bba m_j. Our second idea is to use a distance measure between two bba's m_j and the certain bba m (such that $m(\{ud_i\}) = 1$) representing the bba of the tolerance class X_i relative to the decision value ud_i.

For every ud_i, we define a tolerance class X_i as follows:

$$X_i = \{o_j | dist(m, m_j) < 1 - threshold\}, \, such \, that \, m(\{ud_i\}) = 1 \tag{3.3}$$

Besides, we define a tolerance relation $Tol_{\{ud\}}$ as follows:

$$Tol_{\{ud\}} = U/\{ud\} = \{X_i | ud_i \in \Theta\} \tag{3.4}$$

Remark:

Many distance measures between two bba's were developed (Bauer, 1997; Jousselme et al., 2001; Elouedi et al., 2004; Fixsen & Mahler, 1997; Tessem, 1993; Zouhal & Denœux, 1998) which can be characterized into two kinds:

- Distance measures based on pignistic transformation (Bauer, 1997; Elouedi et al., 2004; Tessem, 1993; Zouhal & Denœux, 1998): For these distances, one unavoidable step is the pignistic transformation of the bba's. Since, there is no bijection between bba's and pignistic probabilities (transformations from the power set to the set). This kind of distance may lose information given by the initial bba's. Besides, we can obtain the same pignistic transformation on two different bba's distributions. So, the distance between the two obtained results does not reflect the actual similarity between the starting bba's distributions.

- Distance measures between bba's defined on the power set were developed in (Jousselme et al., 2001; Fixsen & Mahler, 1997). The first one defines a meaningful metric distance ($dist$) between two bba's m_1 and m_2:

$$dist(m_1, m_2) = \sqrt{\frac{1}{2}(\parallel \vec{m_1} \parallel^2 + \parallel \vec{m_2} \parallel^2 - 2 < \vec{m_1}, \vec{m_2} >)} \tag{3.5}$$

  Where $< \vec{m_1}, \vec{m_2} >$ is the scalar product defined by:

$$< \vec{m_1}, \vec{m_2} > = \sum_{i=1}^{|2^\Theta|} \sum_{j=1}^{|2^\Theta|} m_1(A_i) m_2(A_j) \frac{|A_i \cap A_j|}{|A_i \cup A_j|} \tag{3.6}$$

  with A_i, $A_j \in 2^\Theta$ for i,j= 1..$|2^\Theta|$. $\parallel \vec{m_1} \parallel^2$ is then the square norm of $\vec{m_1}$.

which satisfies the following properties:

1. $Non - negativity: \ dist(m_1, m_2) \geq 0$.
2. $Non - degeneracy: \ dist(m_1, m_2) = 0 \Longleftrightarrow A = B$.
3. $Symmetry: \ dist(m_1, m_2) = dist(m_2, m_1)$.
4. $Triangle\,inequality: \ dist(m_1, m_2) \leq dist(m_1, m_3) + dist(m_3, m_2)$

The second distance measure between two bodies of evidence developed by Fixen and Mahler (Fixsen & Mahler, 1997) is a pseudo-metric, since the condition of non-degeneracy is not respected.

In our work, we choose the distance measure developed in (Jousselme et al., 2001) which is directly defined on bba's and satisfies more properties than the distance measure developed by Fixen and Mahler (Fixsen & Mahler, 1997).

Example 3.4 *Let us continue with the same example to compute the tolerance classes based on the uncertain decision attribute $U/\{ud\}$ by applying the second alternative. The user fixed the value of the threshold at* 0.1*. So, 1- threshold is equal to 0.9. We will obtain the following results:*

Table 3.3: Distance between the bba m_j and the certain bba m

m_j	$dist(m_j, m)$ such that m({yes})=1	$dist(m_j, m)$ such that m({no})=1
m_1	$0.08 < 0.9$	$0.92 > 0.9$
m_2	$1.00 > 0.9$	$0.00 < 0.9$
m_3	$0.66 < 0.9$	$0.34 < 0.9$
m_4	$0.84 < 0.9$	$0.26 < 0.9$
m_5	$1.00 > 0.9$	$0.00 < 0.9$
m_6	$0.93 > 0.9$	$0.07 < 0.9$
m_7	$0.00 < 0.9$	$1.00 > 0.9$
m_8	$0.08 < 0.9$	$0.92 > 0.9$

For the uncertain decision value $ud_1 = yes$, the tolerance class X_1 is equal to $\{o_1, o_3, o_4, o_7, o_8\}$. This tolerance class is obtained according to Table 3.3 which computes the distance between the bba m_j and the certain *bba* m *where* $m(\{yes\}) = 1$.

For the uncertain decision value ud_2=*no, the tolerance class* X_2 *is equal to* $\{ o_2, o_3, o_4, o_5, o_6\}$. *This tolerance class is obtained according to Table 3.3 which also computes the distance between the bba* m_j *and the certain bba* m *where* $m(\{no\}) = 1$.

The relative tolerance relation $Tol_{\{ud\}}$ *is equal to* $U/\{ud\}=\{\{o_1, o_3, o_4, o_7, o_8\}, \{o_2, o_3, o_4, o_5, o_6\}\}$.

Note that in our work, we will focus on the second alternative because the threshold value make the results more flexible. The objects o_1, o_6 and o_8 are near from the certain case and they exist in the right tolerance class.

3.2.3 Set approximation

The set of all objects which can be certainly classified as members of $X \subseteq U$ with respect to $B \subseteq C$ is called *B-lower approximation*. The set of all objects which can be only classified as possible members of X with respect to B is called *B-upper approximation*. The set of all objects which can be decisively classified neither as members of X nor as members of $-X$ with respect to B is called *the boundary region*. If this boundary is non-empty, the set is rough. In this section, we redefine these definitions in the new uncertain context.

To compute the new *lower* and *upper* approximations for our uncertain decision table, we should follow two steps:

1. For each equivalence class based on condition attributes C, combine the bba's of its objects using the mean operator (Murphy, 2000) as follows:

$$\bar{\mathrm{m}}_{[o_j]_C}(E) = \frac{1}{|[o_j]_C|} \sum_{o_i \in [o_j]_C} m_i(E), \; for\, all\, E \subseteq \Theta \tag{3.7}$$

   Let us remember that $[o_j]_C$ is the equivalence class containing the object o_j. In our case, this rule of combination is more suitable to combine these bba's than the rule of combination in eqn. (1.37) which is proposed especially to combine different beliefs on decision for one object and not different bba's relative to different objects.

2. For each tolerance class X_i from $U/\{ud\}$ relative to the uncertain decision attribute ud_i, we compute the new *lower* and *upper* approximations using two alternatives described in the following subsections:

First alternative: restrictive solution

As a first alternative, the *lower* approximation contains all the equivalence classes from U/C included to the tolerance class X_i and have a certain bba.

$$\underline{\mathrm{C}}(X_i) = \{o_j | [o_j]_C \subseteq X_i \text{ and } \bar{\mathrm{m}}_{[o_j]_C}(\{ud_i\}) = 1\} \tag{3.8}$$

We compute the *upper* as the same manner as in the certain case.

$$\bar{\mathrm{C}}(X_i) = \{o_j | [o_j]_C \cap X_i \neq \emptyset\} \tag{3.9}$$

The set of boundary region is defined as follows:

$$BN_C(X_i) = \bar{\mathrm{C}}(X_i) - \underline{\mathrm{C}}(X_i) \tag{3.10}$$

Example 3.5 *We continue with the same example to compute the new lower and upper approximations by applying the first alternative.*

After the first step, we obtain the combined bba for each equivalence classes from U/C using mean operator. Table 3.4 represents the combined bba for the equivalence classes (subsets) $\{o_2, o_6\}$ and $\{o_4, o_5\}$. Note that to simplify the notation, we have used $m_{2,6}$ to mean $\bar{\mathrm{m}}_{[o_2]_C}$.

Table 3.4: Combined bba for the subsets $\{o_2, o_6\}$ and $\{o_4, o_5\}$

$\bar{\mathrm{m}}_{[o_j]_C}$	$m(\{yes\})$	$m(\{no\})$	$m(\Theta)$
m_2	0	1	0
m_6	0	0.95	0.05
$m_{2,6}$	0	0.975	0.025
m_4	0	0.6	0.4
m_5	0	1	0
$m_{4,5}$	0	0.8	0.2

Next, we compute the lower and upper approximations for each tolerance class X_i.

For the uncertain decision value ud_1=yes, let X_1= $\{o_1, o_3, o_4, o_6, o_7, o_8\}$.

Only the subset $\{o_7\}$ is included to X_1 and has a certain bba. Hence, we put it in the lower $\underline{C}X_1$. The subsets $\{o_1\}$, $\{o_3\}$ and $\{o_8\}$ are included to X_1, but they have an uncertain bba. So, we put them in the upper $\bar{C}X_1$. The subsets $\{o_2, o_6\}$ and $\{o_4, o_5\}$ are partially included to X_1. So, we put them in the upper $\bar{C}(X_1)$.

$$\underline{\mathrm{C}}(X_1)=\{o_7\} \text{ and } \bar{\mathrm{C}}(X_1)=\{o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8\}$$

$$BN_C(X_1)=\{o_1, o_2, o_3, o_4, o_5, o_6, o_8\}$$

For the uncertain decision value ud_2=no, let X_2= $\{o_1, o_2, o_3, o_4, o_5, o_6, o_8\}$.

The subsets $\{ o_1\}$, $\{o_2, o_6\}$, $\{o_4, o_5\}$, $\{o_3\}$ and $\{o_8\}$are included to X_2. However, they have an uncertain bba. So, we put them in the upper $\bar{C}(X_2)$.

$$\underline{\mathrm{C}}(X_2)=\emptyset \text{ and } \bar{\mathrm{C}}(X_2)=\{o_1, o_2, o_3, o_4, o_5, o_6, o_8\}$$

$$BN_C(X_2)=\{o_1, o_2, o_3, o_4, o_5, o_6, o_8\}$$

The boundary region is non-empty. So, the uncertain decision Flu is rough.

We can conclude that the first alternative is also restrictive for the set approximation. For example, the subset $\{o_2, o_6\}$ is totally included to X_2 and has a bba near from the certain case. However, we put it in the upper $\bar{\mathrm{C}}X_2$.

Second alternative: flexible solution

As a second alternative, the *lower* approximation contains all the equivalence classes from U/C included to X_i which the distance between the combined bba $\bar{\mathrm{m}}_{[o_j]_C}$ (with $[o_j]_C$ is the equivalence class containing o_j) and the certain bba m (such that $m(\{ud_i\}) = 1$) is less than a *threshold*. In an uncertain context, the *threshold* is needed to give more flexibility to the set approximations.

$$\underline{\mathrm{C}}(X_i) = \{o_j | [o_j]_C \subseteq X_i \, and \, dist(\bar{\mathrm{m}}_{[o_j]_C}, m) \leq threshold\} \tag{3.11}$$

We compute the *upper* as the same manner as in the certain and in the first alternative.

$$\bar{\mathrm{C}}(X_i) = \{o_j | [o_j]_C \cap X_i \neq \emptyset\} \tag{3.12}$$

Note that in the case of uncertainty the *threshold value* gives more flexibility to the tolerance relation and the set approximations. Threshold value is fixed by the user and it should be the same to be coherent.

Example 3.6 *Let us continue with the same example to compute the lower and upper approximations with the second alternative with a threshold fixed at 0.1.*

For the uncertain decision value ud_1=*yes, let* X_1 =$\{o_1, o_3, o_4, o_7, o_8\}$.

The equivalence classes (subsets) $\{o_1\}$, $\{o_3\}$, $\{o_7\}$ *and* $\{o_8\}$ *from* U/C *are included to* X_1. *We should check the distance between their bba* $\bar{m}_{[o_j]_C}$ *and the certain bba* m *(such that* $m(\{yes\}) = 1$*).*

According to Table 3.5, the subsets $\{o_1\}$, $\{o_7\}$ *and* $\{o_8\}$ *are included in the lower* $\underline{C}X_1$. *The subset* $\{o_3\}$ *is included in the upper* $\bar{C}X_1$. *The subset* $\{o_4, o_5\}$ *is partially included to* X_1. *So, we put it in the upper* $\bar{C}X_1$.

$\underline{\mathrm{C}}(X_1)$=$\{o_1, o_7, o_8\}$ and $\bar{\mathrm{C}}(X_1)$=$\{o_1, o_3, o_4, o_5, o_7, o_8\}$

$BN_C(X_1)$=$\{o_3, o_4, o_5\}$

For uncertain decision value ud_2=*no, let* X_2 =$\{ o_2, o_3, o_4, o_5, o_6\}$.

The equivalence classes (subsets) $\{o_2, o_6\}$, $\{o_3\}$ *and* $\{o_4, o_5\}$ *from* U/C *are included to* X_2. *We should check the distance between their bba and the certain bba* m *(such that* $m(\{no\}) = 1$*).*

According to Table 3.5, the subset $\{o_2, o_6\}$ *is included in the lower* $\underline{C}X_2$. *The subsets* $\{o_4, o_5\}$ *and* $\{o_3\}$ *are included to the upper* $\bar{C}X_2$.

$$\underline{C}(X_2)=\{o_2,\, o_6\} \text{ and } \bar{C}(X_2)=\{o_2,\, o_3,\, o_4,\, o_5,\, o_6\}$$

$$BN_C(X_2)=\{o_3,\, o_4,\, o_5\}$$

Table 3.5: Distance between the bba $\bar{m}_{[o_j]_C}$ and the certain bba m

$\bar{m}_{[o_j]_C}$	$dist(\bar{m}_{[o_j]_C}, m)$ *such that* $m(\{yes\}) = 1)$
m_1	$0.07 < 0.1$
m_3	$0.66 > 0.1$
m_7	$0.00 < 0.1$
m_8	$0.07 < 0.1$
$\bar{m}_{[o_j]_C}$	$dist(\bar{m}_{[o_j]_C}, m)$ *such that* $m(\{no\}) = 1)$
$m_{2,6}$	$0.02 < 0.1$
m_3	$0.34 > 0.1$
$m_{4,5}$	$0.26 > 0.1$

Note that in our work, we will also use the second alternative for set approximation like for the tolerance relation due to the threshold value which makes the results more flexible.

3.2.4 Positive region and dependency of attributes

Using the new formalism of *lower* approximation, we can redefine the new positive region denoted by $UPos_C(A, \{ud\})$, is the set of all elements of U that can be uniquely classified to blocks of the partition $U/\{ud\}$, by means of C:

$$UPos_C(A, \{ud\}) = \bigcup_{X_i \in U/\{ud\}} \underline{C}(X_i) \tag{3.13}$$

The uncertain decision attribute ud depends partially on the set of attributes C in a degree $k\,(0 \leq k \leq 1)$, denoted $C \Rightarrow_k \{ud\}$, if

$$k = \gamma_C(A, \{ud\}) = \frac{|UPos_C(A, \{ud\})|}{|U|} \tag{3.14}$$

Example 3.7 *Let us continue with the same example, to compute the positive region and dependency degree of A.*

$$UPos_C(A, \{ud\})=\{o_1,\, o_2,\, o_6,\, o_7,\, o_8\}$$

$$\gamma_C(A, \{ud\}) = \tfrac{5}{8}$$

3.2.5 Belief decision rules

The decision rules induced from our new partially uncertain decision system are denoted belief decision rules where the decision is represented by a bba.

Example 3.8 *Some of the belief decision rules induced from our decision table (see Table 3.1) for the object o_3 and o_7 are as follows :*

'If Headache=yes and Muscle-pain=yes and Temperature=normal then $m_3(\{yes\}) = 0.5; \quad m_3(\Theta) = 0.5$'

'If Headache=no and Muscle-pain=yes and Temperature=very high then $m_7(\{yes\}) = 1$'

Hence, these belief decision rules could be simplified by removing superfluous condition attributes and condition attribute values. With simplification, we can improve the time and the performance of classification for unseen objects. We look to this issue in the next subsections.

3.2.6 Reduct and core

Let us remember that a reduct is a minimal set of attributes from C that preserves the partitioning of the universe and the ability to perform classifications as the whole attribute set C does. In other words, attributes that do not belong to a reduct are superfluous with regard to classification of elements of the universe. The core is the intersection of all possible reducts, it is included in every reduct. Thus, in a sense, the core is the most important subset of attributes, for none of its elements can be removed without affecting the classification power of attributes.

Using the new formalism of positive region, we can redefine these concepts in the new uncertain situation.

Let $c \in C$, the attribute c is *dispensable* in C with respect to ud, if $UPos_C(A, \{ud\}) = UPos_{C-c}(A, \{ud\})$. Otherwise, attribute c is *indispensable* in C with respect to ud.

If all attribute $c \in C$ are *indispensable* in C with respect to ud, then C will be called *independent* with respect to ud.

Let $B \subseteq C$, B is the reduct of C with respect to ud, if B is *independent* set of attributes with respect to ud and:

$$UPos_B(A, \{ud\}) = UPos_C(A, \{ud\}) \tag{3.15}$$

The core is the intersection of all reducts or is the set of all *indispensable* attributes form C with respect to ud.

$$UCore_C(\{ud\}) = \bigcap URed_C(\{ud\}) \tag{3.16}$$

Where $URed_C(\{ud\})$ is the set of all reducts of A relative to ud.

Computing and finding a minimal reduct among all reduct from uncertain decision table is NP-hard problem like in the certain case. In this thesis, we propose a heuristic method of attribute selection. This latter is able to compute reduct in a quick time from a huge number of condition attributes (see Chapter 5).

Example 3.9 *Let us continue with the same example, to compute the possible reducts and the core of A.*

$UPos_{\{Headache\}}(A, \{ud\})=\emptyset$
$UPos_{\{Muscle-pain\}}(A, \{ud\})=\emptyset$
$UPos_{\{Temperature\}}(A, \{ud\})=\{o_1, o_7\}$
$UPos_{\{Headache,\, Muscle-pain\}}(A, \{ud\})=\{o_2,\, o_6\}$
$UPos_{\{Headache,\, Temperature\}}(A, \{ud\})=\{o_1,\, o_2,\, o_6,\, o_7,\, o_8\}$
$UPos_{\{Muscle-pain,\, Temperature\}}(A, \{ud\})=\{o_1,\, o_2,\, o_6,\, o_7,\, o_8\}$

We find that only the subsets {Muscle-pain, Temperature} and {Headache, Temperature} are independent and have the same positive region that the whole set of condition attributes C. So, the subsets {Muscle-pain, Temperature} and {Headache, Temperature} are two reducts relative to the decision Flu in our uncertain decision table. It can be simplified in Table 3.6 or Table 3.7. The only core is the attribute $Temperature$. It is the intersection of the two possible reducts.

Table 3.6: First reduct

U	Muscle-pain	Temperature	Flu
o_1	yes	very high	$m_1(\{yes\}) = 0.95$ $m_1(\{no\}) = 0.05$
o_2	no	high	$m_2(\{no\}) = 1$
o_3	yes	normal	$m_3(\{yes\}) = 0.5$ $m_3(\Theta) = 0.5$
o_4	yes	normal	$m_4(\{no\}) = 0.6$ $m_4(\Theta) = 0.4$
o_5	yes	normal	$m_5(\{no\}) = 1$
o_6	no	high	$m_6(\{no\}) = 0.95$ $m_6(\Theta) = 0.05$
o_7	yes	very high	$m_7(\{yes\}) = 1$
o_8	yes	high	$m_8(\{yes\}) = 1$

Table 3.7: Second reduct

U	Headache	Temperature	Flu
o_1	yes	very high	$m_1(\{yes\}) = 0.95$ $m_1(\{no\}) = 0.05$
o_2	yes	high	$m_2(\{no\}) = 1$
o_3	yes	normal	$m_3(\{yes\}) = 0.5$ $m_3(\Theta) = 0.5$
o_4	no	normal	$m_4(\{no\}) = 0.6$ $m_4(\Theta) = 0.4$
o_5	no	normal	$m_5(\{no\}) = 1$
o_6	yes	high	$m_6(\{no\}) = 0.95$ $m_6(\Theta) = 0.05$
o_7	no	very high	$m_7(\{yes\}) = 1$
o_8	no	high	$m_8(\{yes\}) = 0.9$ $m_8(\Theta) = 0.1$

3.2.7 Value reduct and value core

We need redefine the concept of value reduct and value core for each belief decision rule $R(o_j)$ of the form: **If** $C(o_j)$ **then** m_j as follows:

For all $B \subset C$, Let $X = \{o_k | B(o_j) = B(o_k) \, and \, j \neq k\}$
If $X = \emptyset$ **then** B is a value reduct of $R(o_j)$.
Else If $Max(dist(m_j, \bar{m}_{[o_k]_C})) \leq$ threshold **then** B is a value reduct of $R(o_j)$.

The value core is the intersection of all value reducts for o_j.

$$UCore^j_C(A, \{d\}) = \bigcap URed^j_C(A, \{ud\}) \qquad (3.17)$$

Where $Red_C^j(A, \{ud\})$ is the family of all value reduct of C for o_j with respect to ud.

Example 3.10 *We compute the value reduct of the first decision rule from Table 3.6 with threshold* $= 0.1$*:*

'If Muscle-pain=yes and Temperature=very high then m_1'.

Let us split the rule into:

1. *'If Muscle-pain=yes then m_1' : $X = \{o_3, o_4, o_5, o_7, o_8\}$ and Max(dist(m_1, m_3), dist(m_1, $m_{4,5}$)) $>$ threshold.*

2. *'If Temperature=very high then m_1' : $X = \{o_7\}$ dist(m_1, m_7) $<$ threshold.*

So, Temperature=very high is the only value reduct and the value core for the first rule.

If we compute all the value reducts to Tables 3.6 and 3.7, we obtain respectively Tables 3.8 and 3.9.

Table 3.8: Value reducts 1

U	Muscle-pain	Temperature	Flu
o_1	-	very high	$m_1(\{yes\}) = 0.95$ $m_1(\{no\}) = 0.05$
o_2	no	-	$m_2(\{no\}) = 1$
o_3	yes	normal	$m_3(\{yes\}) = 0.5$ $m_3(\Theta) = 0.5$
o_4	yes	normal	$m_4(\{no\}) = 0.6$ $m_4(\Theta) = 0.4$
o_5	yes	normal	$m_5(\{no\}) = 1$
o_6	no	-	$m_6(\{no\}) = 0.95$ $m_6(\Theta) = 0.05$
o_7	-	very high	$m_7(\{yes\}) = 1$
o_8	yes	high	$m_8(\{yes\}) = 0.9$ $m_8(\Theta) = 0.1$

Table 3.9: Value reducts 2

U	Headache	Temperature	Flu
o_1	-	very high	$m_1(\{yes\}) = 0.95 \;\; m_1(\{no\}) = 0.05$
o_2	yes	high	$m_2(\{no\}) = 1$
o_3	yes	normal	$m_3(\{yes\}) = 0.5 \;\; m_3(\Theta) = 0.5$
o_4	no	normal	$m_4(\{no\}) = 0.6 \;\; m_4(\Theta) = 0.4$
o_5	no	normal	$m_5(\{no\}) = 1$
o_6	yes	high	$m_6(\{no\}) = 0.95 \;\; m_6(\Theta) = 0.05$
o_7	-	very high	$m_7(\{yes\}) = 1$
o_8	no	high	$m_8(\{yes\}) = 0.9 \; m_8(\Theta) = 0.1$

3.3 Generalization Distribution Table (GDT) and Rough Sets (RS)

The rough set theory has attracted attention of many researchers whose contributions have further enhanced it. This theory is not competitive but complementary to other methods and can be often used jointly with other methodologies. In (Dong et al., 1999; Zhong et al., 1998), a soft hybrid induction system for discovering classification decision rules called GDT-RS which is a combination of the *Generalization Distribution Table* (GDT) and the *Rough Sets* (RS) has been proposed. Our second classification technique based on rough sets under the belief function framework namely BRSC-GDT is derived from the belief GDT-RS which is a generalization of the hybrid system GDT-RS in the new context due to its advantages (see subsection 3.3.2).

Before detailing the main characteristics of the GDT-RS system and the belief GDT-RS, we start by the definition of the basics of the *Generalization Distribution Table* (GDT). The latter will not be generalized in the new context. Since, it is based only on the condition attributes.

3.3.1 Generalization Distribution Table (GDT)

The *Generalization Distribution Table* (GDT) is a hypothesis search space for generalization, in which the probabilistic relationship between concepts and instances over discrete domain are presented (Zhong & Ohsuga, 1996). The GDT consists of three components:

1. Possible instances (PI), represented at the top row of GDT, are defined by all possible combinations of attribute values from a database.

2. Possible generalizations of instances (PG), represented by the left column of a GDT, are all possible cases of generalization for all possible instances. A wild card '*' denotes the generalization for instances. For example, the generalization $*b_0c_0$ means that the attribute a is superfluous (irrelevant) for the concept description.

   In other words, if an attribute a takes values from $\{a_0, a_1\}$ and it is superfluous, both $a_0b_0c_0$ and $a_1b_0c_0$ describe the same concept. Therefore, we use the generalization $*b_0c_0$ to describe the set $\{a_0b_0c_0, a_1b_0c_0\}$.

3. Probabilistic relationships between possible instances (PI) and possible generalizations (PG), represented by entries G_{ij} of a given GDT, are defined by means of a probabilistic distribution describing the strength of the relationship between any possible instance and any possible generalization. The prior distribution is assumed to be uniform if background knowledge is not available. Thus, it is defined by:

$$G_{ij} = p(PI_j|PG_i) = \begin{cases} \frac{1}{N_{PG_i}} & \text{if } PG_i \text{ is a generalization of } PI_j \\ 0 & \text{otherwise.} \end{cases} \tag{3.18}$$

where PI_j is the j-th possible instance, PG_i is the i-th possible generalization, and N_{PG_i} is the number of the possible instances satisfying the i-th possible generalization,

$$N_{PG_i} = \prod_{k \in \{l | PG_i[l] = *\}} n_k \tag{3.19}$$

where $PG_i[l]$ is the value of the l-th attribute in the possible generalization PG_i, and n_k is the number of values of the k-th attribute. Certainly, we have $\sum_j G_{ij} = 1$ for any i.

Example 3.11 *Table 3.10 describes a GDT generated from the UDT shown in Table 3.1. To simplify the notations in the GDT, we use H_0 to mean Headache=yes and H_1 to mean Headache=no.*

Table 3.10: Generalization distribution table

PG/PI	$H_0M_0T_0$	$H_0M_0T_1$	$H_0M_0T_2$	$H_0M_1T_0$	$H_0M_1T_1$	$H_0M_1T_2$	$H_1M_0T_0$	....
$*M_0T_0$	1/2						1/2	....
$*M_0T_1$		1/2						....
$*M_0T_2$			1/2					....
$*M_1T_0$				1/2				....
$*M_1T_1$					1/2			....
$*M_1T_2$						1/2		....
$H_0 * T_0$	1/2			1/2				....
$H_0 * T_1$		1/2			1/2			....
$H_0 * T_2$			1/2			1/2		....
$H_1 * T_0$							1/2	....
$H_1 * T_1$								....
$H_1 * T_2$								....
H_0M_0*	1/3	1/3	1/3					....
H_0M_1*				1/3	1/3	1/3		....
H_1M_0*							1/3	....
H_1M_1*								....
$**T_0$	1/4		1/4		1/4		1/4	....
$**T_1$		1/4		1/4		1/4		....
$**T_2$			1/4		1/4		1/4	....
$*M_0*$	1/6	1/6	1/6				1/6	....
$*M_1*$				1/6	1/6	1/6		....
H_0**	1/6	1/6	1/6	1/6	1/6	1/6		....
H_1**							1/6	....

3.3.2 Hybrid system GDT-RS

GDT-RS is a soft hybrid induction system for discovering classification rules from noisy databases (Dong et al., 1999; Zhong et al., 1998). The system is based on a hybridization of the *Generalization Distribution Table* (GDT) and the *Rough Set* (RS) methodology.

The advantages of this hybrid system that can generate, from noisy training data, a set of rules with the minimal description length, having large strength and covering all instances. By using the GDT as a probabilistic search space, unseen instances can be considered in rule discovery process and the uncertainty of a rule, including its ability to predict unseen instances, can be explicitly represented in the strength of the rule.

From the decision table (DT), we can generate decision rules expressed in the following form:

$$P \to Q \text{ with } S$$

- P is a conjunction of descriptors over C.
- Q denotes a concept that the rule describes.
- S is a 'measure of the strength' of the rule.

According to the GDT-RS, the strength S is equal to (Dong et al., 1999; Zhong et al., 1998):

$$S(P \to Q) = s(P) * (1 - r(P \to Q)) \tag{3.20}$$

where $s(P)$ is the strength of the generalization P (the condition of the rule) and r is the noise rate function. The strength of a given rule reflects incompleteness and noise. On the assumption that the prior distribution is uniform, the strength of the generalization $P = PG$ is given by:

$$s(P) = \sum_l p(PI_l|P) = \frac{1}{N_P}|[P]_{DT}| \tag{3.21}$$

where $[P]_{DT}$ is the set of all the objects in DT satisfying the generalization P and N_P is the number of the possible instances satisfying the generalization P which is computed using eqn. (3.19). The strength of the generalization P represents explicitly the prediction for unseen instances. On the other hand, the noise rate is given by:

$$r(P \to Q) = 1 - \frac{|[P]_{DT} \cap [Q]_{DT}|}{|[P]_{DT}|} \tag{3.22}$$

It shows the quality of classification measured by the number of the instances satisfying the generalization P which cannot be classified into class Q. The user can specify an allowed noise level as a threshold value. Thus, the rule candidates with a noise level larger than the given threshold value will be deleted.

3.3.3 Belief GDT-RS

Belief GDT-RS is a generalization of the hybrid system GDT-RS under the belief function framework. The Belief GDT-RS is a soft induction system for discovering classification rules from uncertain database where the uncertainty exists only in the decision attribute values. The latter is represented by the belief functions.

From the uncertain decision table (UDT), we can also generate decision rules expressed in the following form:

$$P \to Q \text{ with } S$$

According to the Belief GDT-RS, the strength S is also equal to the eqn. (3.20). The strength $s(P)$ of the generalization P depends only on the condition attribute values which are certain. So, it is also equal to eqn. (3.21). However, the noise rate function r depends on the condition and the decision attribute values. So, it should be generalized in our uncertain case based on a distance measure as follows:

$$r(P \to Q) = dist(\bar{\mathrm{m}}_{[P]_{DT}}, m), \text{such that } m(\{Q\}) = 1 \qquad (3.23)$$

The idea is to use the distance measure (Jousselme et al., 2001) between two bba's $\bar{\mathrm{m}}_{[P]_{DT}}$ and a certain bba m (such that $m(\{Q\}) = 1$). Where $\bar{\mathrm{m}}_{[P]_{DT}}$ is the combined bba using mean operator (Murphy, 2000) for the equivalence class $[P]_{DT}$ or the set of objects from U satisfying the generalization P. To be more flexible in this uncertain context, if $r(P \to Q) < threshold$. So, $r(P \to Q)$ will be equal to 0.

Example 3.12 *Let us continue with the same UDT shown in Table 3.1 and the GDT shown in Table 3.10. If we have this decision rule: 'If Temperature= very high then Flu=yes'. According to the Belief GDT-RS, the strength S of this decision rule is computed as follows with threshold value equal to 0.1: S= s*(1 - r)=* $\frac{1}{2}$ *with* $s = \frac{1}{4} * 2 = \frac{1}{2}$ *and* $r = dist(m_{1,7}, m) = 0.06 \simeq 0$ *such that m({yes})=1.*

Remark:

The new definitions of the basic concepts of the rough sets and the hyprid system done in the previous sections under the new uncertain context represent a generalization of the certain case.

3.4 Conclusion

In this chapter, we have generalized the basic concepts of rough sets in the new context such as uncertain decision table, tolerance relation, set approximation, positive region and dependency of attributes. These new definitions of the basic concepts relative to rough sets are needed to describe belief rough set classifier. We have also presented the basic concepts of the hybrid system GDT-RS which is a combination of generalization distribution table and rough sets methodology. This hybrid system is also generalized under the belief function framework and is called belief GDT-RS to be used for the creation of the belief rough set classifier based on generalization distribution table.

In the next chapter, we will present the two principal procedures: the construction of our belief rough set classifiers from uncertain data and the classification of new instances.

Chapter 4

Belief classification systems based on rough sets

4.1 Introduction

In this chapter, we detail our two classification approaches based on *Rough Sets* (RS) that are able to learn decision rules from uncertain data. We assume that the uncertainty exists only in the decision attribute values of the *Decision Table* (DT) and is represented by the belief functions. The first classification technique, named *Belief Rough Set Classifier* (BRSC), is only based on the basic concepts of the *Rough Sets* (RS). The second is more sophisticated, called *Belief Rough Set Classifier* based on *Generalization Distribution Table* (BRSC-GDT), and is derived from an hybridization of the *Generalization Distribution Table* and the *Rough Sets* (GDT-RS). The two classifiers aim at simplifying the *Uncertain Decision Table* (UDT) in order to generate significant decision rules for classification process.

In the first part of this chapter, we describe the two principal procedures relative to the BRSC: the construction of the BRSC from uncertain data and the classification of new instances. In fact, the building of the BRSC is based on the redefined basic concepts of rough sets under the belief function framework detailed in the previous chapter which are uncertain decision table, tolerance relation, set approximations, positive region, dependency of attributes and reducts. After the generation of minimal and relevant set of belief decision rules from an uncertain decision table, we can classify unseen instances.

In the second part of this chapter, we describe the construction and the classification procedure relative to the BRSC-GDT. The BRSC-GDT is derived from the hybrid system GDT-RS which is a combination of the *Generalization Distribution Table* (GDT) and the *Rough Sets* (RS) methodology. The soft hybrid system from previous chapter was also redefined in the new context to be called belief GDT-RS. After the generation of the more significant decision rules from an uncertain decision table using BRSC-GDT, we can also classify unseen objects.

4.2 Belief Rough Set Classifier (BRSC)

Our first classification approach based on rough sets under the belief function framework is denoted the *Belief Rough Set Classifier* (BRSC). This technique is only based on the basic concepts of rough sets especially reduct and value reduct redefined in the uncertain context in the previous chapter. It is a new classification method that is able to learn and generate uncertain decision rules from partially uncertain data under the belief function framework. The uncertainty exists only in decision attribute values and is handled through the TBM, one representation of the belief function theory. The set of decision rules generated from belief rough set classifier are able to classify the unseen objects.

The two main procedures used to create the BRSC are as follows:

1. Construction procedure.
2. Classification procedure.

These two procedures are described in the next parts.

4.2.1 BRSC: Construction procedure

The *Belief Rough Set Classifier* (BRSC) seems to learn decision rules from uncertain training dataset. The generated model is able to classify the new objects. To build the BRSC, we propose three main phases. Each phase is explained by examples:

1. Creation of the uncertain training decision table.
2. Simplification of the uncertain training decision table.
3. Generation of the decision rules.

We handle in our work only symbolic condition and decision attribute values. Hence, we do not need the step of discretization detailed in subsection 2.3.1. The Simplification of the uncertain training decision table is the most important phase in the construction procedure.

Phase 1. Creation of the uncertain training decision table

Dividing a data set into a training set and a testing set is a fundamental component in the pre-processing phase of *Data Mining* (DM). Hence, our *Uncertain Decision Table* (UDT) should be divided into two parts: uncertain training decision table (a set of instances used for learning the model generated by the classifier) and uncertain testing decision table (a set of instances used to evaluate the performance of the classifier). We only need the uncertain training decision table to apply the phases (2) and (3). Let us remember that our uncertain decision table (the training set or the testing set) should takes this form: $A = (U, C \cup \{ud\})$ originally described in subsection 3.2.1.

Example 4.1 *Table 4.1 shows an example of uncertain training decision table used to create our classifier. It consists of ten objects, three certain condition attributes* C=$\{Headache,\ Muscle-pain,\ Temperature\}$ *and an uncertain decision attribute* $ud = Flu$ *with two possible values* $\{yes,\ no\}$.

Table 4.1: Uncertain training decision table

U	Headache	Muscle-pain	Temperature	Flu
o_1	no	yes	high	$m_1(\{yes\}) = 0.3 \quad m_1(\Theta) = 0.7$
o_2	no	yes	very high	$m_2(\{yes\}) = 1$
o_3	no	yes	high	$m_3(\{yes\}) = 0.5 \quad m_3(\Theta) = 0.5$
o_4	yes	no	very high	$m_4(\{no\}) = 0.9 \quad m_4(\Theta) = 0.1$
o_5	yes	no	very high	$m_5(\{no\}) = 1$
o_6	no	yes	very high	$m_6(\{yes\}) = 1$
o_7	no	yes	normal	$m_7(\{no\}) = 1$
o_8	yes	yes	very high	$m_8(\{yes\}) = 1$
o_9	no	yes	very high	$m_9(\{yes\}) = 1$
o_{10}	no	yes	normal	$m_{10}(\{no\}) = 1$

The decision rules induced from our training decision table are denoted belief decision rules where the decision is represented by a bba. As an ex-

ample of a rule is : 'If Headache=yes and Muscle-pain=yes and Temperature=normal Then $m_3(\{yes\}) = 0.5$*;* $\quad m_3(\Theta) = 0.5$*'.*

Phase 2. Simplification of the uncertain training decision table

When the uncertain training decision table is ready, we move to the simplification phase. It consists in removing all redundant and unnecessary data for rule discovery. Like in the certain case, the simplification yields more minimal and significant belief decision rules. So, it is the most important phase in the construction procedure.

The simplification phase relative to the BRSC is done by the mean of the following steps:

Step 1. Elimination of the superfluous condition attributes: In uncertain training decision table, there often exist conditional attributes that do not provide (almost) any additional information about the objects. These attributes need to be removed in order to reduce the complexity and cost of decision process. In this step, we remove the superfluous condition attributes that are not in reduct from our uncertain training decision table. This leaves us with a minimal set of attributes that preserve the ability to perform the same classification as the original set of attributes. A decision table may have more than one reduct, and any of these reducts could be used to replace the original table. To achieve this task, we need the new definitions of reduct and core under belief function framework which are made in the previous chapter.

Example 4.2 *Let us continue with the same example in Table 4.1 to compute the possible reducts using the new definition of the concept of reduct using eqn. (3.15).*

$UPos_{\{Headache\}}(A, \{ud\})=\emptyset$
$UPos_{\{Muscle-pain\}}(A, \{ud\})=\{o_4, o_5\}$
$UPos_{\{Temperature\}}(A, \{ud\})=\{o_7\}$
$UPos_{\{Headache, Muscle-pain\}}(A, \{ud\})=\{o_4, o_5, o_8\}$
$UPos_{\{Headache, Temperature\}}(A, \{ud\})=\{o_2, o_6, o_7, o_9, o_{10}\}$
$UPos_{\{Muscle-pain, Temperature\}}(A, \{ud\})=\{o_2, o_4, o_5, o_6, o_7, o_8, o_9, o_{10}\}$
$UPos_{\{Headache, Muscle-pain, Temperature\}}(A, \{ud\})=\{o_2, o_4, o_5, o_6, o_7, o_8, o_9, o_{10}\}$

We find that only the subset {Muscle-pain, Temperature} is independent and has the same positive region that the whole set of condition attributes C. So, the subset {Muscle-pain, Temperature} is the reduct relative to the decision Flu in our uncertain decision table. It can be simplified in Table 4.2. The only core is the subset {Muscle-pain, Temperature}.

Table 4.2: Reduct of uncertain training decision table

U	Muscle-pain	Temperature	Flu
o_1	yes	high	$m_1(\{yes\}) = 0.3 \quad m_1(\Theta) = 0.7$
o_2	yes	very high	$m_2(\{yes\}) = 1$
o_3	yes	high	$m_3(\{yes\}) = 0.5 \quad m_3(\Theta) = 0.5$
o_4	no	very high	$m_4(\{no\}) = 0.9 \quad m_4(\Theta) = 0.1$
o_5	no	very high	$m_5(\{no\}) = 1$
o_6	yes	very high	$m_6(\{yes\}) = 1$
o_7	yes	normal	$m_7(\{no\}) = 1$
o_8	yes	very high	$m_8(\{yes\}) = 1$
o_9	yes	very high	$m_9(\{yes\}) = 1$
o_{10}	yes	normal	$m_{10}(\{no\}) = 1$

Step 2. Elimination of the redundant objects: After removing the superfluous condition attributes from our uncertain training decision table, we will find redundant objects having the same condition attribute values. They may not have the same bba on decision attributes. So, we combine their corresponding bba's using the mean operator (Murphy, 2000) which is the most suitable rule of combination in our context (see subsection 3.2.3).

$$\bar{m}_{[o_j]_B}(E) = \frac{1}{|[o_j]_B|} \sum_{o_i \in [o_j]_B} m_i(E), \; for\, all\, E \subseteq \Theta \tag{4.1}$$

Where B is the reduct of C with respect to ud and $[o_j]_B$ is the equivalence class containing the object o_j with respect to the condition attribute subset B.

Example 4.3 *After removing the superfluous condition attributes and the redundant objects for the uncertain decision table, we obtain Table 4.3. Note that to simplify the notation, we have used $m_{2,6,8,9}$ to mean $\bar{m}_{[o_2]_B}$.*

Table 4.3: Combined objects relative to the reduct

U	Muscle-pain	Temperature	Flu
o_1,o_3	yes	high	$m_{1,3}(\{yes\}) = 0.4 \quad m_{1,3}(\Theta) = 0.6$
o_2,o_6,o_8,o_9	yes	very high	$m_{2,6,8,9}(\{yes\}) = 1$
o_4,o_5	no	very high	$m_{4,5}(\{no\}) = 0.95 \quad m_{4,5}(\Theta) = 0.05$
o_7,o_{10}	yes	normal	$m_{7,10}(\{no\}) = 1$

Step 3. Elimination of the superfluous values of condition attributes: After removing superfluous In this step, we need use the new definition of the concept value reduct for each belief decision rule $R(o_j)$ of the form: **If** $C(o_j)$ **then** m_j (see subsection 3.2.7).

Example 4.4 *If we compute the value reduct of each belief decision rule from Table 4.3 with threshold* $= 0.1$*, we obtain Table 4.4.*

Table 4.4: Simplified belief decision rules

U	Muscle-pain	Temperature	Flu
o_1,o_3	yes	high	$m_{1,3}(\{yes\}) = 0.4 \quad m_{1,3}(\Theta) = 0.6$
o_2,o_6,o_8,o_9	yes	very high	$m_{2,6,8,9}(\{yes\}) = 1$
o_4,o_5	no	-	$m_{4,5}(\{no\}) = 0.95 \quad m_{4,5}(\Theta) = 0.05$
o_7,o_{10}	-	normal	$m_{7,10}(\{no\}) = 1$

Phase 3. Generation of the decision rules

After the simplification of the training uncertain decision table, we can generate short and significant belief decision rules. With simplification, we can improve the time and the performance of classification unseen objects. From one example of UDT, we can find many solutions of simplification where having many reducts and value reducts. The number of solutions, denoted nb, is computed as follows:

$$nb = |URed_C(\{ud\})| * \prod_{j=1}^{n} |URed_C^j(\{ud\})| \tag{4.2}$$

Example 4.5 *Table 4.4 gives the minimal and the unique solution shown as follows:*

1. *'If Muscle-pain=yes and Temperature= high then* $m_{1,3}(\{yes\}) = 0.4$*;* $m_{1,3}(\Theta) = 0.6$*'*
2. *'If Muscle-pain=yes and Temperature= very high then* $m_{2,6,8,9}(\{yes\}) = 1$*'*
3. *'If Muscle-pain=no then* $m_{4,5}(\{no\}) = 0.95$*;* $\quad m_{4,5}(\Theta) = 0.05$*'*
4. *'If Temperature=normal then* $m_{7,10}(\{no\}) = 1$*'*

4.2.2 BRSC: Classification procedure

Once the belief rough set classifier is constructed, the following procedure will be the classification of unseen instances referring to as new objects. Such task is also named the inference task.

Our method is able to ensure the standard classification where each attribute value of the new instance to classify is assumed to be exact and certain.

We search among all belief decision rules which one corresponds to the unseen object. The new instance's decision will be defined by a basic belief assignment. This bba defined on the set on decisions, represents beliefs on the different subsets of classes of the new instance to classify. In order to make a decision and to get the probability of each singular decision, we propose to apply the pignistic transformation.

Example 4.6 *Let us continue with Example 4.5 and assume a new object to classify is characterized by the following values:*

$Headache = no$
$Muscle - pain = no$
$Temperature = high$

This instance corresponding to the third decision rule 'If Muscle-pain=no then $m_{4,5}(\{no\}) = 0.95$*;* $\quad m_{4,5}(\Theta) = 0.05$*'. If we compute the pignistic transformation relative to* $m_{4,5}$*, we find that the decision is no.*

4.3 Belief Rough Set Classifier based on Generalization Distribution Table (BRSC-GDT)

To improve the performance of classification systems based on rough sets, many researchers have extended this theory and have combined it with other methodologies. In (Dong et al., 1999; Zhong et al., 1998) a soft hybrid induction system called GDT-RS for discovering classification rules from databases has been proposed. The system is based on a combination of the *Generalization Distribution Table* and the *Rough Set* methodology.

To pick up the qualities of our *Belief Rough Set Classifier* (BRSC), we combine it with *Generalization Distribution Table* (GDT). The result will be a new classifier called *Belief Rough Set Classifier* based on *Generalization Distribution Table* (BRSC-GDT). The belief GDT-RS which is proposed in the previous chapter is a new definition of the hybrid system GDT-RS under the belief function framework. The advantages of the belief GDT-RS used to build the BRSC-GDT that can generate, from noisy and uncertain training data, a set of rules with the minimal description length, having large strength and covering all instances. Like the BRSC, the uncertainty exists only in decision attributes and is represented by the *Transferable Belief Model* (TBM), one interpretation of the belief function theory. In this section, we detail the two main procedures needed to create the BRSC-GDT: the construction and the classification procedures.

4.3.1 BRSC-GDT: Construction procedure

In this subsection, we explain the construction procedure relative to the BRSC-GDT. This new approach seems to learn a minimal set of decision rules from partially uncertain data having large strength and covering all instances by following three main phases. Each phase is explained by examples:

1. Creation of the uncertain training decision table.
2. Simplification of the uncertain training decision table.
3. Generation of the decision rules.

We can conclude that our two classification systems based on rough sets namely BRSC and BRSC-GDT have the same phases to build their models. The first phase 'the creation of the uncertain training decision table' and the third phase 'the generation of the decision rules' are also similar for both the. However, the main difference between the two classifiers is summarized in the most important phase 'the simplification of the uncertain training decision table'. For each classification system, the steps relative to this latter phase are different. After the simplification of the uncertain decision table which is based on the reduction of the superfluous condition attributes (reduct) and the superfluous condition attribute values (value reduct), the BRSC generates all possible decision rules. However, the BRSC-GDT, in addition to of the simplification of the uncertain decision tables, it eliminates the contradictory decision rules. Finally, if one decision rule has more than one possibility of simplification, we keep only the decision rule with best strength. So, the BRSC-GDT can generate smallest models than the BRSC. The BRSC-GDT can also be more fast than the BRSC by avoiding some iterations.

Phase 1. Creation of the uncertain training decision table

Let us remember that this first phase consists of preparing a training uncertain decision table, where the structure is described in subsection 3.2.1, used to build the model using phases (2) and (3).

Example 4.7 *Let us continue with the same example of uncertain training decision table shown in Table 4.1 to illustrate the idea.*

Phase 2. Simplification of the uncertain training decision table

The simplification of the uncertain decision table consists of removing all redundant and unnecessary data for rule discovery process. The phase of simplification relative to the BRSC-GDT is done by the means of the following steps:

Step 1. Creation of the GDT: The standard GDT depends only on condition attributes and not on decision attribute values. In the uncertain context, our GDT still have the same structure as in (Zhong & Ohsuga, 1996). Note that this step can be omitted because the prior distribution of a generalization can be calculated using eqns. (3.18) and (3.19).

Step 2. Definition of the compound object: Objects from U having the same condition attribute values are considered as one object over C, called the compound object o'_j and defined as follows:

$$o'_j = \{o_i | C(o_i) = C(o_j)\} \tag{4.3}$$

For objects composing each compound object, combine their bba's using the mean operator as follows:

$$m'_j(E) = \frac{1}{|o'_j|} \sum_{o_j \in o'_j} m_j(E), \forall E \subseteq \Theta \tag{4.4}$$

The reasons for choosing the mean operator (Murphy, 2000) to combine these bba's are the same as described in the previous chapter (see subsection 3.2.3).

Example 4.8 *Let us continue with the same example in Table 4.1. By applying the step 2, we obtain the following tables:*

Table 4.5: Compound objects

U	Headache	Muscle-pain	Temperature
o'_1 (o_1, o_3)	no	yes	high
o'_2 (o_2, o_6, o_9)	no	yes	very high
o'_4 (o_4, o_5)	yes	no	very high
o'_7 (o_7, o_{10})	no	yes	normal
o'_8 (o_8)	yes	yes	very high

Table 4.6: Combined bba's

U	ud
o'_1	$m'_1(\{yes\}) = 0.4$ $m'_1(\Theta) = 0.6$
o'_2	$m'_2(\{yes\}) = 1$
o'_4	$m'_4(\{no\}) = 0.95$ $m'_4(\Theta) = 0.05$
o'_7	$m'_7(\{no\}) = 1$
o'_8	$m'_8(\{yes\}) = 1$

Step 3. Elimination of the contradictory compound objects: For any compound object o'_j from U, compute $r_{ud_i}(o'_j)$ which representing a noise rate for each decision value ud_i, If there exists a ud_i such that $r_{ud_i}(o'_j)$ $= min\{r_{ud_{i'}}(o'_j)|ud_{i'} \in \Theta\} < T_{noise}$ (threshold value), then we assign the decision value ud_i to the object o_j. If there is no $ud_i \in \Theta$ such that $r_{ud_i}(o'_j)$ $< T_{noise}$, we treat the compound object o'_j as a contradictory one, and set the decision value of o'_j to $\perp$(uncertain).

According to the belief GDT-RS, the noise rate is calculated using eqn. (3.23). This latter is used in this step for detecting the contradictory compound objects. So, we propose that the noise rate in this step which is also based on a distance measure is reformulated as follows:

$$r_{ud_i}(o'_j) = dist(m'_j, m), \text{such that } m(\{ud_i\}) = 1 \tag{4.5}$$

The idea is to use the distance between two bba's m'_j and a certain bba m (such that $m(\{ud_i\}) = 1$). With this manner, we can check that the decisions of all instances belong to the compound object are near from a certain case. So, it is considered as a not contradictory compound object.

Note that the distance measure used in this step is the same as used and described in Chapter 3, i.e. the Jousselme distance (Jousselme et al., 2001).

Example 4.9 *We fixed the threshold value of T_{noise} at 0.05. For the compound object o'_1, the noise rate for the decision value ud_1 is equal to $r_{ud_1}(o'_1)$=0.36 and the noise rate for the decision value ud_2 is equal to $r_{ud_2}(o'_1)$=0.63. So, o'_1 is a contradictory compound object. By applying the step 3 to Tables 4.5 and 4.6, we obtain Table 4.7:*

Table 4.7: Contradictory and not contradictory compound objects

U	Headache	Muscle-pain	Temperature	ud
o'_1	no	yes	high	$\perp$
o'_2	yes	yes	very high	yes
o'_4	no	no	very high	no
o'_7	no	yes	normal	no
o'_8	yes	yes	very high	yes

Step 4. Minimal description length of decision rule: Select from U', the set of all the compound object except the contradictory ones, one compound object o'_j then create a discernibility vector (the row or the column with respect to o'_j in the discernibility matrix (Skowron & Rauszer, 1992)) for o'_j (see subsection 2.2.7).

The discernibility matrix of A is a symmetric n*n matrix with entries a_{ij} as given below. Each entry thus consists of the set of attributes upon which objects o_i and o_j differ.

$$a_{ij} = \{c \in C | c(o_i) \neq c(o_j)\} \; and \; ud(o_i) \neq ud(o_j) \;\; for \, i, j = 1, ..., n \quad (4.6)$$

Next, we compute all the so-called local relative reducts for the compound object o'_j by using the discernibility function $f_A(o'_j)$. It is a boolean function of k boolean variables corresponding to the k condition attributes defined as below:

$$f_A(o'_j) = \wedge\{\vee a_{ij} | 1 \leq i \leq n, a_{ij} \neq \emptyset\} \quad (4.7)$$

The set of all prime implicants of $f_A(o'_j)$ determines the sets of all reduct values of the compound object o'_j.

Example 4.10 *According to Table 4.7, the discernibility vector for the compound object o'_2 is shown in Table 4.8.*

Table 4.8: Discernibility vector for o'_2

U'	$o'_1(\perp)$	$o'_2(yes)$	$o'_4(no)$	$o'_7(no)$	$o'_8(yes)$
$o'_2(yes)$	Temperature	$\emptyset$	Headache, Muscle-pain	Temperature	$\emptyset$

Compute all the so-called local relative reducts for object o'_2 by using the discernibility function, we obtain two value reducts namely {Headache, Temperature} and {Temperature, Muscle-pain} where $f_A(o'_2)$=$(Temperature) \wedge (Headache \vee Muscle{-}pain) \wedge (Temperature) = (Headache \wedge Temperature) \vee (Muscle - pain \wedge Temperature)$.

Step 5. Selection of the best rules: We start by constructing the decision rules obtained from the local reduct for the object o'_j and revising their strength according to the belief GDT-RS and using the eqn. (3.20). Then, we select only the best rule for o'_j having minimal description length, large strength or covering instances as much as possible.

Example 4.11 *For example, the following rules are acquired for object o'_2:*

1. *'If Headache=no and Temperature=very high then yes' with $S = 1 * \frac{1}{2} = 0.5$.*
2. *'If Muscle-pain=yes and Temperature=very high then yes' with $S = 2 * \frac{1}{2} = 1$.*

The rule 'If Muscle-pain=yes and Temperature=very high then yes' is selected for the instance o'_2 according to it strength.

Step 6. Stopping criterion: Let $U' = U' - \{o'_j\}$. $If\ U' \neq \emptyset$, then go back to *Step* 4. Otherwise, STOP.

Phase 3. Generation of the decision rules

After the simplification of the uncertain decision table based on BRSC-GDT, new version of the GDT-RS induction system under the belief function framework, we can generate only one subset of decision rules with minimal description length, having large strengths and covering all instances. The induced decision rule has the form described as below:

$$\alpha \longrightarrow \beta \text{ with } S$$

-α denotes the conjunction of the some conditions that a concept must satisfy.
-β denotes a concept that the rule describes.
-S is a $measure\ of\ strength$ of which the rule holds.

Remark:

The decision rules generated from the two classifiers: BRSC and BRSC-GDT have not the same structure. However, the classification of unseen objects is possible for the two cases.

Example 4.12 *The following table gives the set of decision rules generated from the chosen database shown in Table 4.1*:

Table 4.9: Decision rules

U'	rules	strengths
o'_2, o'_8	If Temperature=very high and Muscle-pain=yes then yes	1
o'_4	If Muscle-pain=no then no	0.167
o'_7	If Temperature=normal then no	0.25

4.3.2 BRSC-GDT: Classification procedure

If the model generated by the BRSC-GDT is ready, we can classify a new objects characterized by certain attribute values. This step is similar as the certain case (see subsection 2.3.2).

Remark:

The time complexity of our new solutions namely BRSC and BRSC-GDT is the same which is O(kn^2Nr_{max}), where:

- n is the number of instances in a given database,
- k stands for the number of attributes,
- Nr_{max} is the maximal number of reducts for instances.

These algorithms are not suitable for large databases with many number of attributes. We suggest application of a heuristic method for attribute selection in pre-processing stage before using our BRSC and BRSC-GDT (see Chapter 5).

4.4 Conclusion

In this chapter, we have described the two main procedures namely the construction and the classification procedures relative to our two proposed classifiers BRSC and BRSC-GDT. Both are shown to simplify the uncertain decision table and to generate more efficient decision rules for the classification process. The uncertainty appears only in decision attribute values which is represented through the belief functions. We have detailed the major phases needed to build them under the belief function framework. The most important phase is the simplification of the uncertain training decision table. The classifiers are able to generate a minimal and a significant set of uncertain decision rules to classify unseen objects.

In the next chapter, we suggest to improve the qualities of our two approaches in term of time complexity of learning by applying a heuristic method for attribute selection and improve the classification accuracy by applying the notion of dynamic reduct.

Chapter 5

Improvements of the belief rough set classifiers

5.1 Introduction

In the previous chapter, we have detailed our two classification approaches based on rough sets under the belief function framework namely the *Belief Rough Set Classifier* (BRSC) and the *Belief Rough Set Classifier* based on *Generalization Distribution Table* (BRSC-GDT). In this chapter, we propose to improve the qualities of our two approaches in term of time complexity of learning by applying a heuristic method for attribute selection and improve the accuracy of classification by applying the notion of dynamic reduct.

In the first part of this chapter, we detail our heuristic method for feature selection based on rough sets under the uncertain context to avoid a costly calculation of the reduct from the uncertain decision table. The objective is to reduce the time requirement needed to construct the two approaches: the BRSC and the BRSC-GDT. Using this heuristic method to construct them results in two other versions of classifiers denoted by H-BRSC and H-BRSC-GDT.

In the second part of this chapter, we redefine the concept of dynamic reduct under the belief function framework to avoid unstable results. The objective is to obtain more accurate and stable classification. Using the notion of dynamic reduct to construct them leads to two other versions of classifiers denoted by D-BRSC and D-BRSC-GDT.

5.2 Heuristic method for attribute selection

The exhaustive research to compute all possible reducts needed for the simplification of the uncertain decision table is an expensive solution which is only practical for simple datasets. In fact, finding all possible reducts is NP-hard complexity problem yields costly calculation. Especially with large datasets containing huge numbers of features, which would be impossible to process further. Another problem of finding all possible reducts using rough sets: What is the best reduct for the classification process? Which one we should select? The solution to these problems is to apply a heuristic attribute selection method.

In this section, we propose a heuristic method of feature selection based on rough sets that is on one hand able to select the relevant features and on the other hand does not damage the performance of induction, from our partially uncertain data under the belief function framework without hard calculation. At first, we give an overview among the heuristic attribute selection methods based on rough sets. Then, we choose one of them based on its advantages to generalized it in the uncertain context.

Our solution could be applied for our both proposed classification approaches BRSC and BRSC-GDT to obtain two other versions of classification systems denoted by H-BRSC and H-BRSC. We carry experimentations on real-world databases to evaluate the performance of this heuristic to reduce the learning time (see Chapter 6).

5.2.1 Heuristic attribute selection methods using rough sets

Recently feature selection has received considerable attention from machine learning and knowledge discovery researchers interested in improving the performance of their algorithms and in cleaning their data. In handling large databases, feature selection is even more important since many learning algorithms may falter or take too long to run before the data are reduced.

There are several attempts to solve this problem by rough set community. Most feature selection methods based on rough set can be grouped into two categories: exhaustive or heuristic search of an optimal set of attributes.

1. Exhaustive feature selection methods based on rough sets (Pawlak, 1991; Skowron & Rauszer, 1992): They consist of selecting the optimal

reduct from all possible reducts. However, most of time only one subset of attributes is used to reduce a dataset. So, all the calculation involved in discovering the rest are pointless (see subsection 2.2.6).

2. Heuristic feature selection methods based on rough sets (Chouchoulas & Shen, 2001; Jensen & Shen, 2003; Modrzejewski, 1993; Wroblewski, 1995; Zhong et al., 2001): They carry out an exhaustive search to find a minimal combination of features that is sufficient to construct a hypothesis consistent with a given set of examples.

In this section, we present an overview of these heuristic methods in order to adapt one of them to extract the more relevant subset of attributes in a quick time from our partially uncertain data under the belief function framework.

QuickReduct algorithm

QuickReduct algorithm (adapted from (Chouchoulas & Shen, 2001)) attempts to calculate a reduct without exhaustively generating all possible subsets. It starts off with an empty set and adds in turn, one at a time, those attributes that result in the greatest increase in the rough set dependency metric. According to the QuickReduct algorithm, the dependency of each attribute is calculated and the best candidate is chosen. This process continues until the dependency of the reduct equals the consistency of the dataset (1 if the dataset is consistent).

Drawbacks: Determining the consistency of the entire dataset is reasonable for most datasets. However, it may be infeasible for very large data, so alternative stopping criteria may have to be used. One such criterion could be to terminate the search when there is no further increase in the dependency measure.

ReverseReduct algorithm

Other developments include ReverseReduct, adapted from (Chouchoulas & Shen, 2001), where the strategy is backward elimination of attributes as opposed to the current forward selection process. Initially, all attributes appear in the reduct candidate; the least informative ones are incrementally removed until no further attribute can be eliminated without introducing inconsistencies.

Drawbacks: However, ReverseReduct is not guaranteed to find a minimal subset. Using the dependency function to discriminate between candidates may lead the search down a non-minimal path. It is impossible to predict which combinations of attributes will lead to an optimal reduct based on changes in dependency with the addition or deletion of single attributes. They do result in a close-to-minimal subset.

PRESET algorithm

Another approach to generating reducts from decision table have been developed in (Modrzejewski, 1993), called the PRESET algorithm. It is another feature selector that uses rough set theory to rank heuristically the features.

Drawbacks: Since PRESET algorithm does not try to explore all combinations of the features. It is certain that PRESET algorithm fails on problems whose attributes are highly correlated. In feature selection, a good subset of features is one whose features are highly correlated with the decision.

Attribute selection heuristic based on genetic algorithm

In (Wroblewski, 1995), another attribute selection heuristic was proposed which uses genetic algorithms to discover optimal or close-to-optimal reducts. Reduct candidates are encoded as bit strings, with the value in position i set if the ith attribute is present. The fitness function depends on two parameters. The first parameter is the number of bits set. The function penalizes those strings which have larger numbers of bits set, driving the process to find smaller reducts. The second is the number of classifiable objects given this candidate. The reduct should discern between as many objects as possible (ideally all of them).

Drawbacks: Although, this approach is not guaranteed to find minimal subsets, it may find many subsets for any given dataset. It is useful for situations where new objects are added to or old objects are removed from a dataset. The reducts generated previously can be used as the initial population for the new reduct-determining process. The main drawback is the time taken to compute each bit strings fitness.

Attribute selection heuristic using ant-based framework

In (Jensen & Shen, 2003), authors propose a new rough set approach to feature selection based on *Ant colony optimization* (ACO). The latter has been applied to many combinatorial problems. It is particularly attractive for feature selection since ants can discover the best feature combinations as they traverse the graph. The precomputed heuristic desirability of edge traversal is the entropy measure, with the subset evaluation performed using the rough set dependency heuristic. The number of ants used is set to the number of features, with each ant starting on a different feature. Ants construct possible solutions until they reach a rough set reduct. To avoid fruitless searches, the size of the current best reduct is used to reject those subsets whose cardinality exceeds this value.

Drawbacks: However, this heuristic is very complex with many parameters. Besides, the major drawbacks of this approach is the complexity: the problem space is represented as a graph of all features and this does not scale well to very large datasets.

Attribute selection heuristic using filter-based approach

In (Zhong et al., 2001), a heuristic filter-based approach is presented based on rough set theory. The proposed algorithm starts with the core of the dataset (those attributes that cannot be removed without introducing inconsistencies) and incrementally adds attributes based on a heuristic measure (strategy for attribute selection). Additionally, a threshold value is required as a stopping criterion to determine when a reduct candidate is ″near enough″ to being an optimal reduct. On each iteration, those objects that are consistent with the current reduct candidate are removed. It is an optimization which can speed up the algorithm on each iteration as it removes objects that are already in the positive region.

Advantages: The advantages of this heuristic are as follows:

1. It is fast with time complexity is equal to $O(k * n)$. Where n represents the number of instances and k is the number of the condition attributes.

2. It generates only one reduct. The latter is a minimal combination of features that is sufficient to reduce the decision table without affecting the classification power.

3. It is better than the other heuristic methods by trying to avoid the local optimum. It is due to the efficient strategy for attribute selection.

4. It holds more flexibility with threshold.

5. It is not complex with many parameters.

5.2.2 Heuristic method for attribute selection under belief function framework

Among all these heuristic methods for feature selection based on rough sets, we choose the heuristic proposed in (Zhong et al., 2001) to adapt it in our uncertain context due to its advantages.

Notations

The following notations are used to introduce the algorithm:

- R: the set of selected condition attributes,
- P: the set of unselected condition attributes,
- ε: reduct threshold.

Principle

Our algorithm uses the attributes from core as an initial attribute subset. Next, it selects attributes one by one from unselected ones using some strategies, and adds them to the attribute subset until a reduct approximation is obtained.

The strategy for attribute selection used in this algorithm can be described as follows: select a given attribute c, if by adding it to the subset R of attributes, the cardinality of $UPos_{R\cup\{c\}}(A, \{ud\})$ increases faster and the $max{-}size(UPos_{R\cup\{c\}}(A, \{ud\})\ /R\cup\{c\}\cup\{ud\}))$ and $max{-}size(U/R\cup\{c\})$ are larger than by adding any other attribute. The discussed conditions can be competitive. So, we choose in our quality criterion the result of multiplication of the three values.

Algorithm

Initial state
(1) R $\leftarrow UCore_C(A, \{ud\})$ // calculation of the core
$P \leftarrow C - UCore_C(A, \{ud\})$
(2) **while** $(\gamma_R(A, \{ud\}) < \varepsilon)$
(3) U $\leftarrow U - UPos_R(\{ud\})$ //optimization
(4) $\forall\ c \in P$
(5) $v_c = |UPos_{R\cup\{c\}}(A, \{ud\})|$
(6) $m_c = max - size(UPos_{R\cup\{c\}}(A, \{ud\})/R \cup \{c\} \cup \{ud\}))$
(7) $x_c = max - size(U/R \cup \{c\})$
(8) Choose c with largest $v_c * m_c * x_c$
(9) $R \leftarrow R \cup \{c\}$
(10) P$\leftarrow$ P -{c}
(11) **return** R

Example 5.1 *Let us take Table 5.1 to compute the reduct using our heuristic method. This latter contains eight objects, three certain condition attributes C={Headache, Muscle-pain, Temperature} and an uncertain decision attribute ud=Flu with possible value {yes, no} representing* Θ*.*

Table 5.1: Uncertain decision table 2

U	Headache	Muscle-pain	Temperature	Flu
o_1	yes	yes	very high	$m_1(\{yes\}) = 1$
o_2	yes	no	high	$m_2(\{no\}) = 1$
o_3	yes	yes	normal	$m_3(\{yes\}) = 0.5 \quad m_3(\Theta) = 0.5$
o_4	no	yes	normal	$m_4(\{no\}) = 0.6 \quad m_4(\Theta) = 0.4$
o_5	no	yes	normal	$m_5(\{no\}) = 1$
o_6	yes	no	high	$m_6(\{no\}) = 1$
o_7	no	yes	very high	$m_7(\{yes\}) = 1$
o_8	no	yes	high	$m_8(\{yes\}) = 1$

We start by computing the core (the set of indispensable condition attributes) as follows:

Remove the $'Headache'$ *attribute from the condition attributes:*

$UPos_{\{Muscle-pain,Temperature\}}(A, \{ud\})$=$\{o_1, o_2, o_6, o_7, o_8\}$= $UPos_C(A, \{ud\})$
So, the $'Headache'$ *attribute is not indispensable.*

Remove the $'Temperature'$ *attribute from the condition attributes:*
$UPos_{\{Headache,Muscle-pain\}}(A, \{ud\})$=$\{o_8\} \neq UPos_C(A, \{ud\})$
So, the $'Temperature'$ *attribute is indispensable.*

Remove the $'Muscle-pain'$ *attribute from the condition attributes:*
$UPos_{\{Headache,Temperature\}}(A, \{ud\})$=$\{o_1, o_2, o_6, o_7, o_8\} = UPos_C(A, \{ud\})$
So, the $'Muscle-pain'$ *attribute is not indispensable.*

Only the $Temperature$ *attribute is indispensable with respect to* ud*. Hence, it is the core.*

We have, in the initial state:
R=$UCore_C(A, \{ud\})$=$\{Temperature\}$
P=*C*-$UCore_C(A, \{ud\})$=*{Headache, Muscle-pain}*

Setting reduct threshold: $\varepsilon = \gamma_C(A, \{ud\})$=*5/8, the termination condition will be* $\gamma_R(A, \{ud\}) \geq 5/8$*. Since* $\gamma_R(A, \{ud\})= 2/8 < 5/8$*, R is not a reduct, and we must continue adding other condition attributes to R until a reduct is obtained.*

The positive region of the attribute {Temperature} with respect to {ud}: $UPos_{\{Temperature\}}(A, \{ud\})= \{o_1, o_7\}$*. The initial state is* $U = \{o_2, o_3, o_4, o_5, o_6, o_8\}$*, without consistent objects* $\{o_1, o_7\}$*, is shown in Table 5.2.*

Table 5.2: Initial state

U	Temperature	Flu
o_2	high	$m_2(\{no\}) = 1$
o_3	normal	$m_3(\{yes\}) = 0.5 \quad m_3(\Theta) = 0.5$
o_4	normal	$m_4(\{no\}) = 0.6 \quad m_4(\Theta) = 0.4$
o_5	normal	$m_5(\{no\}) = 1$
o_6	high	$m_6(\{no\}) = 1$
o_8	high	$m_8(\{yes\}) = 1$

Next, we have two candidates $Muscle - pain$ and $Headache$. Tables 5.3 and 5.4 give the results of adding $Muscle - pain$ and $Headache$ to R, respectively.

From Tables 5.3 and 5.4, we obtain the following positive regions:
$UPos_{\{Muscle-pain,Temperature\}}(A, \{ud\}) = \{o_2, o_6, o_8\}$
$UPos_{\{Headache,Temepature\}}(A, \{ud\}) = \{o_2, o_6, o_8\}$
$v_{Muscle-pain}$=$|UPOS_{\{Muscle-pain,Temperature\}}(A, \{ud\})|$=3
$v_{Headache}$=$|UPOS_{\{Headache,Temepature\}}(A, \{ud\})|$=3

The two candidates have the same v_c. So, we should check the value of m_c
$UPos_{\{Muscle-pain,Temperature\}}(A, \{ud\})/\{Muscle-pain, Temerature, Flu\} = \{\{o_2, o_6\}, \{o_8\}\}$
$UPos_{\{Headache,Temperature\}}(A, \{ud\})/\{Headache, Temperature, Flu\} = \{\{o_2, o_6\}, \{o_8\}$
$m_{Muscle-pain}$=2
$m_{Headache}$=2

The two candidates have the same m_c. So, we should check the value of x_c
$U/\{Muscle - pain, Temperature\}$=$\{\{o_2, o_6\}, \{o_3, o_4, o_5\}, \{o_8\}\}$
$U/\{Headache, Temperature\}$=$\{\{o_2, o_6\}, \{o_3\}, \{o_4, o_5\}, \{o_8\}\}$
$x_{Muscle-pain} = 3$
$x_{Headache} = 2$

One can see that by selecting the $'Muscle-pain'$ or $'Headache'$ attribute, we can reduce the number of contradictory instances. Since the maximal set is in $U/\{Muscle - pain, Temperature\}$. Then, according to our selection strategies, $Muscle-pain$ should be selected first. After adding $Muscle-pain$ to R, $\gamma_R(A, \{ud\})$= 5/8 $\geq$ 5/8. The process is finished. Thus, the selected attribute subset is $\{Muscle - pain, Temperature\}$.

Table 5.3: Selecting 'Muscle-pain' attribute

U	Muscle-pain	Temperature	Flu
o_2	no	high	$m_2(\{no\}) = 1$
o_3	yes	normal	$m_3(\{yes\}) = 0.5 \quad m_3(\Theta) = 0.5$
o_4	yes	normal	$m_4(\{no\}) = 0.6 \quad m_4(\Theta) = 0.4$
o_5	yes	normal	$m_5(\{no\}) = 1$
o_6	no	high	$m_6(\{no\}) = 1$
o_8	yes	high	$m_8(\{yes\}) = 1$

Table 5.4: Selecting 'Headache' attribute

U	Headache	Temperature	Flu
o_2	yes	high	$m_2(\{no\}) = 1$
o_3	yes	normal	$m_3(\{yes\}) = 0.5 \quad m_3(\Theta) = 0.5$
o_4	no	normal	$m_4(\{no\}) = 0.6 \quad m_4(\Theta) = 0.4$
o_5	no	normal	$m_5(\{no\}) = 1$
o_6	yes	high	$m_6(\{no\}) = 1$
o_8	no	high	$m_8(\{yes\}) = 1$

5.3 Dynamic method for attribute selection

Feature selection is an important pre-processing stage in machine learning. Rough set theory provides an attractive mechanism for feature selection (Modrzejewski, 1993; Pawlak, 1991; Skowron & Rauszer, 1992). The simplest approach is based on the calculation of reduct. Another issue in real-world applications is the uncertain, imprecise or incomplete data.

However, computing reducts from uncertain and noisy data make the results unstable, and sensitive to the sample data. All of these limit the application of rough set theory. Dynamic reducts (Bazan et al., 1994) can lead to better performance in very large datasets, and also provide the ability to accommodate noisy data. The rules calculated by means of dynamic reducts are better predisposed to classify unseen cases, because these reducts are in some senses the most stable reducts, and they appear most frequently in sub-decision systems created by random samples of a given decision system.

In this section, we will generalize the concepts of dynamic reduct and dynamic core under belief function framework. Before that, we start by defining the concepts of dynamic reduct and dynamic core in the certain case.

5.3.1 Dynamic reduct and dynamic core

If $A = (U, C \cup \{d\})$ is a decision table, then any system $B = (U', C \cup \{d\})$ such that $U' \subseteq U$ is called a subtable of A. Let F be a family of subtables of A (Bazan et al., 1994).

$$DR(A, F) = RED(A, d) \cap \bigcap_{B \in F} RED(B, d) \tag{5.1}$$

Any element of $DR(A, F)$ is called an F-dynamic reduct of A. From the definition of dynamic reducts, it follows that a reduct of A is dynamic if it is also a reduct of all subtables from a given family F. This notation can be sometimes too restrictive so we apply a more general notion of dynamic reduct. They are called (F, ε)-dynamic reducts, where $0 \leq \varepsilon \leq 1$. The set $DR_\varepsilon(A, F)$ of all (F, ε)-dynamic reducts is defined by

$$DR_\varepsilon(A, F) = \left\{ R \in RED(A, d) : \frac{|\{B \in F : R \in RED(B, d)\}|}{|F|} \geq 1 - \varepsilon \right\} \tag{5.2}$$

Nevertheless, computing reducts from uncertain and noisy data leads to results which are unstable and sensitive to the sample data. Therefore it is important to search the most stable reduct denoted dynamic reduct (Bazan et al., 1994) or computing reduct containing dynamic core defined as follows:

If $A = (U, C \cup \{d\})$ is a decision table, then any system $A' = (U', C \cup \{ud\})$ such that $U' \subseteq U$ is called a subtable of A. Let F be a family of subtables of A.

$$DCore_C(A, F) = Core_C(A, \{ud\}) \cap \bigcap_{A' \in F} Core_C(A', \{ud\}) \tag{5.3}$$

Any element of $DCore_C(A, F)$ is called an F-dynamic core of A. From the definition of dynamic core, it follows that a core of A is dynamic if it is also a core of all subtables from a given family F.

5.3.2 Dynamic reduct and dynamic core under belief function framework

Our decision system is characterized by high level of uncertain and noisy data. One of the issues with such a data is that the resulting reducts are not stable, and are sensitive to sampling. The belief decision rules generated are not suitable for classification. The solution to this problem is to redefine the concept of dynamic reduct in the new context. The rules calculated by means of dynamic reducts are better predisposed to classify unseen objects, because they are the most frequently appearing reducts in sub-decision systems created by random samples of a given decision system. In this subsection, we will generalize the concepts of dynamic reduct and dynamic core in the uncertain context. The objective is to extract more stable reducts from the uncertain decision system.

Using the new definition of reduct in our uncertain context, we can redefine the concept of dynamic reduct as follows:

$$UDR(A,\, F) = URED(A, ud) \cap \bigcap_{B \in F} URED(B,\, ud) \tag{5.4}$$

Where F be a family of subtables of A. This notation can be sometimes too restrictive so we apply a more general notion of dynamic reduct. They are called $(F,\, \varepsilon)$-dynamic reducts, where $0 \leq \varepsilon \leq 1$. The set $UDR_\varepsilon(A,\, F)$ of all $(F,\, \varepsilon)$-dynamic reducts is defined by:

$$UDR_\varepsilon(A,\, F) =$$

$$\left\{ R \in\, URED(A,\, ud) : \frac{|\{B \in F : R \in URED(B,\, ud)\}|}{|F|} \geq 1 - \varepsilon \right\} \tag{5.5}$$

If $A = (U,\, C \cup \{ud\})$ is an uncertain decision table, then any system $A' = (U',\, C \cup \{ud\})$ such that $U' \subseteq U$ is called a subtable of A. Let F be a family of subtables of A.

$$UDCore_C(A,\, F) = UCore_C(A,\, \{ud\}) \cap \bigcap_{A' \in F} UCore_C(A',\, \{ud\}) \tag{5.6}$$

Any element of $UDCore_C(A,\, F)$ is called an F-dynamic core of A. From the definition of dynamic core, it follows that a core of A is dynamic if it is also a core of all subtables from a given family F.

Example 5.2: *To compute the dynamic reduct of the uncertain decision system A. We divide our uncertain decision system into two subtables B and B' to obtain a family F of sub-decision system. B contains the objects o_1, o_2, o_3, o_4 and B' contains the objects o_5, o_6, o_7, o_8. The two subtables have the same reducts as the whole decision system A. So, the subsets {Headache, Temperature} and {Muscle-pain, Temperature} are dynamic reducts relative to the chosen family F.*

5.4 Conclusion

In this chapter, we have presented two major ideas to improve the qualities of our two classification approaches namely BRSC and BRSC-GDT. The first idea focuses on reducing the time complexity requirement to construct our two classifiers by applying a heuristic feature selection method in a pre-processing stage. Using this heuristic method leads to two other versions of classifiers denoted by H-BRSC and H-BRSC-GDT.

The second idea attempts to increase the classification accuracy of the generated rules of our two classifiers by applying the notion of dynamic reduct to reduce the uncertain decision table. This dynamic method results in two other versions of classifiers denoted by D-BRSC and D-BRSC-GDT.

In the next chapter, we will carry experimentations to evaluate the performance of our two classifiers and their versions on modified real-world databases by artificially introducing uncertainty in the decision attribute values and on a naturally uncertain web usage mining database.

Chapter 6

Simulation and experimental results

6.1 Introduction

Implementation of our classification systems based on rough sets under the belief function framework seems imperative since it allows us to have an idea concerning the feasibility of our solutions and its efficiency (see Appendix A). Once the different programs are implemented, we have performed several tests and simulations on real-world databases obtained from the U.C.I. repository[1] for checking the feasibility of our approaches, judging their qualities and comparing between them. The latters were artificially modified in order to include uncertainty in decision attribute. To further evaluate our two belief classification techniques and their versions, we have also performed experimentations on a naturally uncertain web usage database (Lingras & West, 2004). This dataset was obtained from web access logs of the introductory computing science course at Saint Mary's University. The chosen evaluation criteria for this experimental part are the time requirement of the construction procedure, the size of the models and the classification accuracy.

Different results from these simulations have been compared and analyzed in order to evaluate our classifiers. We have also made comparisons with the results given by a similar classifier namely, the *Belief Decision Tree* (BDT) (Elouedi et al., 2001; Trabelsi et al., 2006, 2007) which is more suitable since it deals with the same hypotheses (see subsection 6.2.5).

[1] http://www.ics.uci.edu/ mlearn/MLRepository.html

The first part of this chapter presents all preliminary notions for setting experimentations like the description of the experimental databases, the evaluation criteria, the testing strategy and finally, we give a description of our comparative classifier: BDT. We further report the different results obtained from the artificial uncertain databases and the naturally uncertain web usage database in order to demonstrate the applicability and the effectiveness of our classification systems based on rough sets under belief function framework in comparison with belief decision tree.

6.2 Experimental settings

In this section, we provide all necessary parts related to the evaluation of the belief rough set classifier approaches: BRSC, BRSC-GDT, H-BRSC, H-BRSC-GDT, D-BRSC, D-BRSC-GDT. We start by providing a description of the used databases. Then, three evaluation criteria are proposed for checking the performance of our solutions. Next, the notion of cross validation is described which is our testing strategy. Finally, we give an overview of the belief decision tree which is the existing comparative classifiers used to more evaluate the performance of our approaches.

6.2.1 Description of databases

U.C.I repository databases

As mentioned in the beginning of this chapter, we have performed simulations on real-world databases obtained from the U.C.I repository of machine learning databases.

A brief description of these nominal-valued databases is presented in Table 6.1 where # instances, #attributes and #decision values denote respectively the total number of instances in the database, the number of condition attributes and the number of decision attributes values. Since we only deal with symbolic attribute values, we have chosen the following databases:

Table 6.1: Description of databases

Database	#instances	#attributes	#decision values
W. Breast Cancer	690	8	2
Balance Scale	625	4	3
C. Voting records	497	16	2
Zoo	101	17	7
Nursery	12960	8	3
Solar Flares	1389	10	2
Lung Cancer	32	56	3
Hayes-Roth	160	5	3
Car Evaluation	1728	6	4
Lymphography	148	18	4
Spect Heart	267	22	2
Tic-Tac-Toe Endgame	958	9	2

Despite the great interest to the field of data mining from uncertain data these last years, real uncertain data and more specifically, with uncertain decision attribute values represented through belief functions are not very available. Hence, most researchers on the field of uncertain data mining and in order to make experimentations create uncertainty in artificial way.

Data sets of the U.C.I. repository (and consequently those given in Table 6.1) are crisp data and do not contain neither uncertain attributes nor uncertain decision values. Thus, we should contaminate these datasets by creating for each instance of each dataset, a belief distribution on the different possible decision values labels in order to get the same structure of the training set needed in building belief rough set classifiers. Namely, each instance will keep its original attribute values unchanged but its decision value will be replaced by a bba.

Constructing uncertainty: The belief rough set classifiers are essentially built from decision tables characterized by uncertain decision attribute where the uncertainty is represented by a bba given on the set of the possible decision values. So, the question is how will we construct these bba's?

These bba's are created artificially. They take into account three basic parameters:

- The real decision values of the training instances
- Degree of uncertainty
  - No uncertainty (certain case): we take P = 0
  - Low degree of uncertainty: we take $0 < \text{P} \leq 0.3$
  - Middle degree of uncertainty: we take $0.3 < \text{P} \leq 0.6$
  - High degree of uncertainty: we take $0.6 < \text{P} \leq 1$
- The number N of bba's composing the bba on instance's class.

For each object, we build N bba's and combine them conjunctively using eqn (1.37). The resulting bba is the bba describing our belief about the actual decison value to witch the object belongs. Each bba has almost 2 focal elements:

1. The first is the actual decision value ud_i of the object with bba, $m(\{ud_i\}) = 1 - P$ (P is a probability generated randomly).
2. The second is a subset θ of Θ (generated randomly) such that the actual decision value of the object under consideration belongs to θ and every of the other decision values belongs to θ with probability P. $m(\theta) = P$

A larger P gives a larger degree of uncertainty.

Real web usage database

Besides of the modified U.C.I repository databases used in this chapter, we have also performed experimentations on a real and an uncertain web usage database (see a part in Appendix B). The uncertainty exists in the decision attribute values and represented via the belief functions. The latter were obtained from web access logs of the introductory computing science course at Saint Mary's University. The course is 'Introduction to Computing Science and Programming' offered in the first term of the first year. The initial number of students in the course was 180. It is reduced over the course of the semester to 140 students. The students in the course come from a wide variety of backgrounds, such as computing science major hopefuls, students taking the course as a required science course, and students taking the course as a science or general elective. As is common in a first year course, students' attitudes towards the course also vary a great deal. Lingras and West (2004) showed that visits from students attending the first course could fall into one of the following three categories (decision values):

- Studious: These visitors download the current set of notes. Since they download a limited/current set of notes, they probably study class-notes on a regular basis.
- Crammers: These visitors download a large set of notes. This indicates that they have stayed away from the class-notes for a long period of time. They are planning for pretest cramming.
- Workers: These visitors are mostly working on class or lab assignments or accessing the discussion board.

Data preparation: Data quality is one of the fundamental issues in data mining. Poor data quality always leads to poor quality of results. Data preparation is an important step before applying data mining algorithms. The data preparation consisted of two phases: data cleaning and data transformation.

1. Data cleaning involved removing hits from various search engines and other robots. Some of the outliers with large number of hits and document downloads were also eliminated. This reduced the data set by 5%.
2. Data transformation required the identification of web visits (Lingras & West, 2004). Certain areas of the web site were protected, and the users could only access them using their IDs and passwords. The activities in the restricted parts of the web site consisted of submitting a user profile, changing a password, submission of assignments, viewing the submissions, accessing the discussion board, and viewing current class marks. The rest of the web site was public. The public portion consisted of viewing course information, a lab manual, class-notes, class assignments, and lab assignments. If users only accessed the public web site, their IDs would be unknown. Therefore, web users were identified based on their IP address. This also made sure that the user privacy was protected. A visit from an IP address started when the first request was made from the IP address. The visit continued as long as the consecutive requests from the IP address had sufficiently small delay.

The web logs were preprocessed to create an appropriate representation of each user, corresponding to a visit. The abstract representation of a web user is a critical step that requires a good knowledge of the application domain. Previous personal experience with the students in the course suggested that some of the students print preliminary notes before a class and an updated copy after the class. Some students view the notes on-line on a regular basis. Some students print all the notes around important days such as midterm and final examinations. In addition, there are many accesses on Tuesdays and Thursdays, when in-laboratory assignments are due. On and off-campus points of access can also provide some indication of a user's objectives for the visit. Based on some of these observations, it was decided to use the following attributes for representing each visitor (Lingras & West, 2004):

- On campus/Off campus access.
- Day time/Night time access: 8 a.m. to 8 p.m. were considered to be the daytime.
- Access during lab/class days or non-lab/class days: All the labs and classes were held on Tuesdays and Thursdays. The visitors on these days are more likely to be workers.
- Number of hits.
- Number of class-notes downloads.

The first three attributes had binary values of 0 or 1. However, the last two variables represent the number of hits and number of class-notes were integer values. As detailed in the previous chapters that our classification systems handle only symbolic attributes. Hence, the two last attributes (the number of hits and number of class-notes) are discretized using the hybrid system called RSBR proposed in (Nguyen, 1997) (see subsection 2.3.1).

Total visits were 23754, the visits where no class-notes were downloaded were eliminated, since these visits correspond to either casual visitors or workers. Elimination of outliers and visits from the search engines further reduced the sizes of the data set to 7965.

Instead of representing an object as belonging to a cluster ud_i (decision value). It also associated by an expert a degree of belief (bba m_j) in the object belonging to the cluster ud_i with $\Theta = \{ud_i | 1 \leq i \leq k\}$ be the set of all the clusters (the possible decision values).

6.2.2 Evaluation criteria

The relevant criteria used to judge the performance of our proposed classifiers in both construction and classification procedures are as follows:

-To evalute the construction procedure for each classifier, we choose:

- **The time requirement** which represents the number of seconds needed to construct each classifier. So, this criterion is used to find the more fasters classifiers in building phase.

-To evalute the classification procedure for each classifier, we choose:

- **The size** which is characterized by the number of the decision rules generated from each classifier. It is generally accepted that the fewer terms in a model the better is. This criterion can also be used to find the more fasters approaches in classification phase due to the small size of their models.

- **The classification accuracy** which is determined by measuring the number of instances it, correctly, classifies among the total number of testing instances presented to the classifier. Hence, this criterion is used to find the more accurate approaches for classify the new objects.

We will use the *Percent of Correct Classification* (PCC) as a performance indicator. The latter represents the percent of the correct classification of the testing instances which are classified according to the induced decision rules. It is given by:

$$PCC = \frac{number\,of\,well\,classified\,instances}{total\,number\,of\,classified\,instances} * 100 \tag{6.1}$$

The PCC is computed as follows: for each testing instance, we make comparison between its real decision value and the decision value given by the classification procedure of BRSC or BRSC-GDT. Hence, the number of well classified instances represents the number of testing instances for which the decision value obtained by the classifier is the same as their real decision values.

A PCC equal to 100% qualifies the belief rough set classifier as an excellent classifier, whereas a 'null' classifier has a PCC equal to 0%.

An equivalent criterion, also used in the literature, measure the proposition of incorrectly classified instances. This is known as the *error rate* (r=1-PCC). We can also mention the Kappa statistics (Siegel & Castellan, 1988) which penalizes the PCC by the percentage of instances that could be correctly classified by chance.

6.2.3 Testing strategy

The construction of the belief rough set approaches requires a training set for building models. We need also a testing set for evaluating the performance of the classifiers. The data set is divided into k parts. k-1 partitions are used as the training set to build the classifier, the last is used as the testing set to evaluate it. The procedure is repeated k times, each time another part is chosen as the testing set. The k results are then averaged. The advantage of this method is that all instances are used for both training and testing, and each instance is used for testing exactly once. This method, called a cross-validation, permits an efficient estimation of the evaluation criteria. 10-fold cross-validation is commonly used.

6.2.4 Technique of sampling

The technique for sampling of uncertain decision table and dynamic reduct computation in the uncertain context is similar with the technique employed in the certain case (Bazan et al., 1994) which consists of the following steps.

- In the first step, some of samples of the uncertain decision table are computed randomly. Next, reducts for all these subtables are calculated.
- In the second step, reducts with the stability coefficients higher than a fixed threshold are extracted. They are considered as dynamic reducts.

In our experimentations, we take five samples of the uncertain training decision table to avoid the costly calculation of the dynamic reduct:

- Sample 1 contains 10% of the training set
- sample 2 contains 20% of the training set
- Sample 3 contains 30% of the training set
- Sample 4 contains 40% of the training set
- sample 5 contains 50% of the training set

6.2.5 Belief decision tree: comparative classifier

To more evaluate the performance of our classifiers, we compare the obtained results with those obtained from another similar classifier. We choose the *Belief Decision Tree* (BDT) approach (Denœux & Skarstein-Bjanger, 2000; Elouedi et al., 2001; Vannoorenberghe & Denœux, 2002). BDT is a decision tree in an uncertain environment where the uncertainty is represented through the TBM. The uncertainty appears in the actual decision attribute values of training objects. The model generated by the BDT is belief decision rules. Hence, the comparison between the two classifiers is possible. The following parts, give us an overview about BDT.

Description

A belief decision tree is a decision tree in an uncertain environment where the uncertainty is represented by the TBM. In our work, we focus on BDT proposed in (Elouedi et al., 2001) where there are two approaches of building. These latters deal with only symbolic attributes.

- The averaging approach is an extension of the classical approach developed by Quinlan and based on the gain ratio criterion (Quinlan, 1993).
- The conjunctive approach represented ideas behind the TBM itself and based on a distance criterion.

Belief decision tree parameters

In this part, we define the major parameters leading to the construction of the belief decision tree where objects may have uncertain decision attribute values.

Parameter 1. Attribute selection measures: The major parameter ensuring the building of a decision tree is the attribute selection measure allowing to determine the attribute to assign to a node of the induced BDT at each step.

Averaging Approach: Under this approach, the attribute selection measure is based on the entropy computed from the average pignistic probabilities computed from the pignistic probabilities of each instance in the node. The following steps are proposed to choose the appropriate attribute:

1. Compute the pignistic probability $BetP_j$ of each object o_j by applying the pignistic transformation to m_j.

2. For each $ud_i \in \Theta$, compute the average pignistic probability function $BetP\{S\}$ taken over the set of objects S.

$$BetP\{S\}(\{ud_i\}) = \frac{1}{|S|} \sum_{o_j \in S} BetP_j(\{ud_i\}) \tag{6.2}$$

3. Compute the entropy $Info(S)$ of the average pignistic probabilities in the set S. This $Info(S)$ value is equal to:

$$Info(S) = -\sum_{i=1}^{s} BetP\{S\}(\{ud_i\}) \log_2 BetP\{S\}(\{ud_i\}) \tag{6.3}$$

4. Select an attribute c. Collect the subset S_v^c made with cases of S having v as a value for the attribute c. Then, compute the average pignistic probability for objects in subset S_v^c. Let the result be denoted by $BetP\{S_v^c\}$.

5. Compute $Info_c(S)$:

$$Info_c(S) = \sum_{v \in D(c)} \frac{|S_v^c|}{|S|} Info(S_v^c) \tag{6.4}$$

   where $D(c)$ is the domain of the possible values of the attribute c and $Info(S_v^c)$ is computed using $BetP\{S_v^c\}$.

6. Compute the information gain provided by the attribute c in the set of objects S such that:

$$Gain(S, c) = Info(S) - Info_c(S) \tag{6.5}$$

7. Using the Split Info, compute the gain ratio relative to attribute c:

$$Gain\,Ratio(S, c) = \frac{Gain(S, c)}{Split\,Info(S, c)} \tag{6.6}$$

   Where

$$Split\,Info(S, c) = -\sum_{v \in D(c)} \frac{|S_v^c|}{|S|} \log_2 \frac{|S_v^c|}{|S|} \tag{6.7}$$

8. Repeat the same process for every attribute c belonging to the set of attributes that can be selected. Next, choose the one that maximizes the gain ratio.

Conjunctive Approach: It is based on an intra-group distance quantifying each attribute value how strongly objects are close from each others. The different steps upon this attribute selection measure ensuring the building of a decision tree are the following ones:

1. For each training object, compute from the bba m_j:

$$K_j(E) = -\ln q_j(E)\,\forall E \subseteq \Theta \tag{6.8}$$

2. For each attribute value v of an attribute c, compute the joint $K\{S_v^c\}$ defined on Θ, the set of possible classes by:

$$K\{S_v^c\}(E) = \sum_{o_j \in S_v^c} K_j(E) \tag{6.9}$$

3. For each attribute value, the intra-group distance SumD(S_v^c) is defined by:

$$SumD(S_v^c) = \frac{1}{|S_v^c|} \sum_{o_j \in S_v^c} \sum_{E \subseteq \Theta} (K_j(E) - \frac{1}{|S_v^c|} K\{S_v^c\}(E))^2 \tag{6.10}$$

4. Compute $SumD_c(S)$ representing the weighted sum of the different $SumD(S_v^c)$ relative to each value v of the attribute c:

$$SumD_c(S) = \sum_{v \in D(c)} \frac{|S_v^c|}{|S|} SumD(S_v^c) \tag{6.11}$$

5. By analogy to our averaging approach, we may also compute $Diff(S,c)$ defined as the difference between $SumD(S)$ and $SumD_c(S)$:

$$Diff(S,c) = SumD(S) - SumD_c(S) \tag{6.12}$$

Where

$$SumD(S) = \frac{1}{|S|} \sum_{o_j \in S} \sum_{E \subseteq \Theta} (K^{\Theta}\{o_i\}(E) - \frac{1}{|S|} K^{\Theta}\{S\}(E))^2 \tag{6.13}$$

6. Using the Split Info (see eqn. (6.7)), compute the diff ratio relative to the attribute c:

$$Diff\ Ratio(S, c) = \frac{Diff(S, c)}{Split\ Info(S, c)} \tag{6.14}$$

7. For every attribute repeat the same process, and choose the one that maximizes the diff ratio.

Parameter 2. Partitioning strategy: The partitioning strategy for the construction of a belief decision tree is similar to the partitioning strategy used in the standard tree. Since we deal with only symbolic attributes, we create an edge for each value of the attribute chosen as a decision node.

Parameter 3. Stopping criteria: Four strategies are proposed as stopping criteria:

1. If the treated node includes only one instance.

2. If the treated node includes only instances for which the m_j's are equal.

3. If all the attributes are split.

4. If the value of the applied attribute selection measure (using either the gain ratio or the diff ratio) for the remaining attributes is less or equal than zero.

Parameter 4. Structure of leaves: Each leaf in the induced tree will be characterized by a bba. Using the averaging approach, the leaf's bba is equal to the average of the bba's of the objects belonging to this leaf. However, in the conjunctive approach, the leaf's bba is the result of combination of the bba's of objects belonging to this leaf using the conjunctive rule.

Belief decision trees procedures

The BDT is composed of two principal procedures: the building of the tree from training objects with uncertain classes and the classification of new instances.

1. **Building procedure:** Building a decision tree in this context of uncertainty will follow the same steps presented in C4.5 algorithm. Furthermore, this algorithm is generic since it offers two possibilities for selecting attributes by using either the averaging approach or the conjunctive one.

2. **Classification procedure:** Once the belief decision tree is constructed, it is able to classify an object described by an exact value for each one of its attributes, we have to start from the root of the belief decision tree, and repeat to test the attribute at each node by taking into account the attribute value until reaching a leaf. As a leaf is characterized by a bba on classes, the pignistic transformation is applied to get the pignistic probability on the classes of the object to classify in order to decide its class. For instance, one can choose the class having the highest pignistic probability.

   Belief decision trees also deal with the classification of new instances characterized by uncertainty in the values of their attributes. The idea to classify such objects is to look for the leaves that the given instance may belong to by tracing out possible paths induced by the different attribute values of the object to classify. The new instance may belong to many leaves where each one is characterized by a basic belief assignment. These bba's are combined using disjunctive rule of combination in order to get beliefs on the instance's classes.

Pruning belief decision tree

Inducing a BDT may lead in most cases to very large trees with bad classification accuracy and difficult comprehension. As in standard decision trees within belief decision trees, pre-pruning and post-pruning methods have been proposed to cope with this problem.

Pre-pruning method has been developed in (Elouedi et al., 2002) by improving the stopping criteria concerning the value of the selection measure in BDT using a discounting factor and in (Denœux & Skarstein-Bjanger, 2000) where impurity measure, based on evidence-theoretic uncertainty measure, is used to grow the tree and has the advantage to define simultaneously the pruning strategy. It allows to control the complexity of the tree, thus avoiding overtraining.

In (Trabelsi et al., 2007), one of standard post-pruning methods, namely the minimal cost-complexity pruning has been adapted in averaging and conjunctive approaches in order to simplify the belief decision tree and improve its classification accuracy.

6.3 Results and analysis

The objectives to carry experimentations on the chosen real-world databases are:

1. Evaluate our two proposed approaches *Belief Rough Set Classifier* (BRSC) and *Belief Rough Set Classifier* based on *Generalization Distribution Table* (BRSC-GDT).

2. Check the performance of our heuristic feature selection method for reducing the time requirement of the construction procedure relative to BRSC and BRSC-GDT by testing their versions namely H-BRSC and H-BRSC-GDT.

3. Check the performance of our dynamic reduct approach for improving the quality of classification relative to BRSC and BRSC-GDT by testing their versions namely D-BRSC and D-BRSC-GDT.

4. Compare the obtained results with those given by another similar classifier, namely the *Belief Decision Tree* (BDT) approach developed in (Elouedi et al., 2001).

Note that pruning step is indispensable to improve the size and the accuracy of the BDT. In (Trabelsi et al., 2007), we have concluded that the post-pruning is better than pre-pruning (Elouedi et al., 2002) in terms of size and accuracy. For these reasons, we focus in our experimentations only on the post-pruned belief decision tree in averaging and conjunctive approaches. We took best results between the two approaches related to BDT.

6.3.1 Experimental results from U.C.I repositry databases

In this part, we report the experimental results from U.C.I repositry databases shown in Table 6.1 by applying all classification approaches (pruned BDT, BRSC, BRSC-GDT, H-BRSC, H-BRSC-GDT, D-BRSC, D-BRSC-GDT) based on three evaluation criteria: time requirement, size of models and classification accuracy. To more illustrate the idea, we give for each evaluation criteria three figures relative to three databases.

Time requirement criterion

We start by the first evaluation criterion representing the time requirement needed to build our classification approaches. Table 6.2 gives the different results obtained from all databases where the different classifiers are sorted from the best to the worst. Note that the time requirement is almost the same for the different degrees of uncertainty. In Table 6.2, we only report the mean of the different degrees of uncertainty (no uncertainty, low uncertainty, middle uncertainty and high uncertainty).

Table 6.2: Experimental results for the time requirement (seconds)(mean)

Bases	H-BRSC-GDT	H-BRSC	BRSC-GDT	BRSC	Pruned BDT	D-BRSC-GDT	D-BRSC
W. Breast Cancer	41	66	136	154	156	381	393
Balance Scale	35	42	126	129	139	282	297
C. Voting records	27	68	106	110	117	224	235
Zoo	34	39	96	101	103	207	221
Nursery	127	191	356	380	386	867	1106
Solar Flares	104	108	146	157	160	391	412
Lung Cancer	15	29	42	48	56	174	187
Hayes-Roth	28	34	82	91	93	198	205
Car Evaluation	105	133	156	178	189	418	426
Lymphography	43	65	97	102	108	209	217
Spect Heart	52	71	89	109	111	226	233
Tic-Tac-Toe	98	107	127	139	149	293	312

From this table, we find that the H-BRSC-GDT is faster to construct than all approaches in all databases. For example, the time requirement for

C. Voting records database is only 27s. This positive result is due to the heuristic feature selection step which can compute sufficient solution without costly calculation and to the hybrid system GDT-RS which can avoid many iterations by eliminating some unnecessary decision rules in the discovery process like the contradictory decision rules.

The table also shows that our heuristic feature selection method has a very good effect on reducing the time requirement to build both the classifiers: BRSC and BRSC-GDT. So, the H-BRSC and H-BRSC-GDT become very faster than BRSC and BRSC-GDT. For example, the time requirement for *Nursery* database goes from 380s with BRSC to 191s with H-BRSC.

However, the application of the notion of dynamic reduct increase the time requirement to construct both the classifiers: BRSC and BRSC-GDT. So, the D-BRSC and the D-BRSC-GDT need more cost calculation than the BRSC and the BRSC-GDT. For example, the time requirement for *Zoo* database goes from 101s with BRSC to 221s with D-BRSC. Hence, we can see that D-BRSC-GDT and D-BRSC are the worst. This negative result is due to the dynamic reduct method which compute the more stable reduct from many subtables obtained from the uncertain decision table.

The table also shows that the *Generalization Distribution Table* (GDT) methodology has a very good effect on reducing the time requirement. So, the BRSC-GDT, the H-BRSC-GDT and the D-BRSC-GDT are respectively faster than the BRSC, the H-BRSC and the D-BRSC. For example, the time requirement for *Spect Heart* database goes from 109s with BRSC to 89s with BRSC-GDT.

Besides, we also find that our classification systems based on rough sets namely H-BRSC-GDT, H-BRSC, BRSC-GDT, BRSC are more fast than the post-pruned BDT. This result is also explicable where the step of pruning increases the time requirement needed to build the BDT.

To illustrate the results, Figures 6.1, 6.2 and 6.3 give a graphical presentation for the time requirement for all approaches relative to some databases namely *W. Breast Cancer*, *C. Voting records* and *Lung Cancer*.

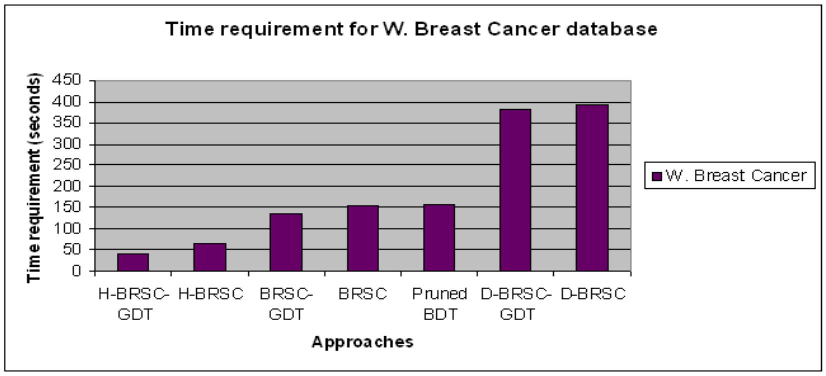

Figure 6.1: Time requirement for W. Breast Cancer database

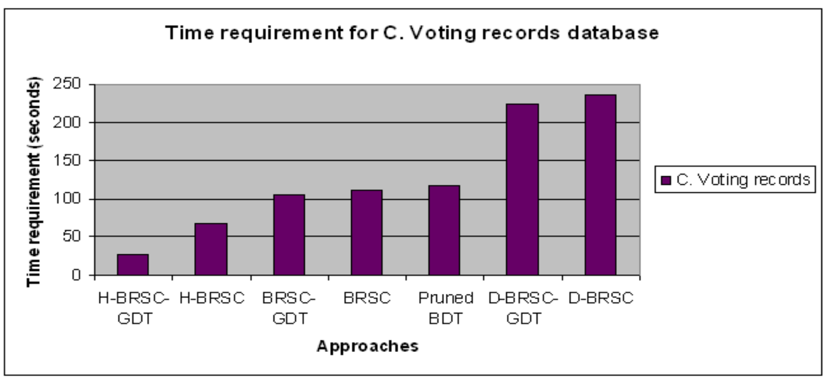

Figure 6.2: Time requirement for C. Voting records database

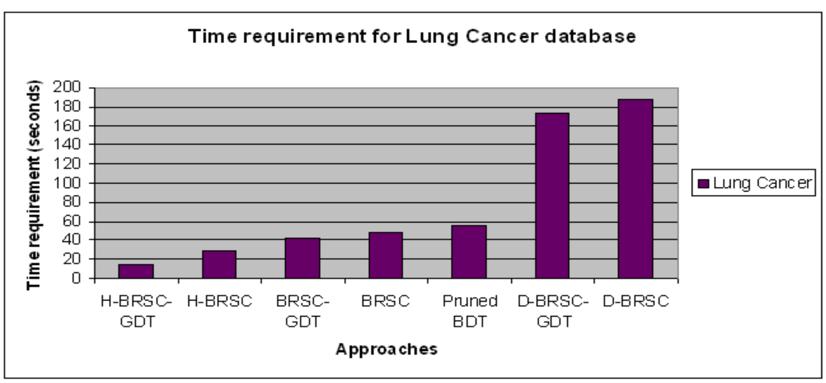

Figure 6.3: Time requirement for Lung Cancer database

Size criterion

We move to the second evaluation criterion representing the number of the learned decision rules generated from BDT, BRSC, BRSC-GDT, H-BRSC, H-BRSC-GDT, D-BRSC, D-BRSC-GDT. From Table 6.3 to Table 6.6, we detail the different results relative to the chosen databases for the different degree of uncertainty where the different classifiers are sorted from the best to the worst. Table 6.3 gives the results relative to the certain case (no uncertainty). Tables 6.4, 6.5 and 6.6 present the results relative to the uncertain case (low uncertainty, middle uncertainty and high uncertainty). Table 6.7 summarizes the mean size for the different degrees of uncertainty from all databases relative to our classifiers.

Table 6.3: Experimental results for the size (certain case)

Bases	BRSC-GDT	H-BRSC-GDT	D-BRSC-GDT	Pruned BDT	BRSC	H-BRSC	D-BRSC
W. Breast Cancer	35	36	38	40	48	50	51
Balance Scale	40	43	43	45	59	60	62
C. Voting records	32	32	34	35	47	49	51
Zoo	22	22	22	26	35	35	35
Nursery	178	180	183	191	210	214	219
Solar Flares	94	104	108	109	110	125	128
Lung Cancer	19	19	19	21	22	22	22
Hayes-Roth	27	27	29	30	37	39	40
Car Evaluation	150	152	158	166	171	174	176
Lymphography	33	35	35	44	56	58	59
Spect Heart	37	39	39	41	49	52	52
Tic-Tac-Toe	98	101	103	105	120	121	121

Table 6.4: Experimental results for the size (low uncertainty)

Bases	BRSC-GDT	H-BRSC-GDT	D-BRSC-GDT	Pruned BDT	BRSC	H-BRSC	D-BRSC
W. Breast Cancer	37	38	40	43	49	50	51
Balance Scale	48	49	50	50	57	61	62
C. Voting records	35	36	36	38	50	52	53
Zoo	25	27	28	28	36	43	44
Nursery	182	186	187	197	223	225	228
Solar Flares	96	109	116	117	127	127	128
Lung Cancer	21	22	22	23	26	28	28
Hayes-Roth	28	29	30	33	38	39	41
Car Evaluation	151	157	160	170	174	177	179
Lymphography	38	40	41	48	57	60	57
Spect Heart	38	40	41	43	51	52	52
Tic-Tac-Toe	102	104	104	109	122	122	125

Table 6.5: Experimental results for the size (middle uncertainty)

Bases	BRSC-GDT	H-BRSC-GDT	D-BRSC-GDT	Pruned BDT	BRSC	H-BRSC	D-BRSC
W. Breast Cancer	39	40	42	47	50	52	53
Balance Scale	52	53	53	55	59	61	63
C. Voting records	40	41	42	43	51	54	55
Zoo	28	29	33	34	39	44	45
Nursery	185	188	189	199	229	232	234
Solar Flares	101	114	119	120	122	127	129
Lung Cancer	22	23	25	25	29	30	30
Hayes-Roth	28	29	30	33	38	39	41
Car Evaluation	156	159	164	170	175	178	179
Lymphography	43	44	45	50	58	60	61
Spect Heart	43	44	45	47	55	55	56
Tic-Tac-Toe	107	111	112	124	125	126	126

Table 6.6: Experimental results for the size (high uncertainty)

Bases	BRSC-GDT	H-BRSC-GDT	D-BRSC-GDT	Pruned BDT	BRSC	H-BRSC	D-BRSC
W. Breast Cancer	43	44	46	52	53	54	55
Balance Scale	53	55	56	56	61	62	64
C. Voting records	48	49	49	52	56	57	59
Zoo	31	32	35	40	42	46	47
Nursery	207	211	219	222	246	248	251
Solar Flares	113	123	127	128	130	130	132
Lung Cancer	22	25	27	29	30	31	31
Hayes-Roth	31	33	36	37	42	46	49
Car Evaluation	159	161	165	173	175	181	182
Lymphography	52	52	56	57	58	59	61
Spect Heart	44	44	45	51	57	58	58
Tic-Tac-Toe	118	119	119	127	127	127	127

Table 6.7: Experimental results for the size (mean)

Bases	BRSC-GDT	H-BRSC-GDT	D-BRSC-GDT	Pruned BDT	BRSC	H-BRSC	D-BRSC
W. Breast Cancer	38	39	41	45	50	51	52
Balance Scale	48	50	51	51	59	61	63
C. Voting records	39	40	40	42	51	53	54
Zoo	26	28	30	32	38	42	44
Nursery	188	191	194	202	227	229	233
Solar Flares	101	112	117	121	110	127	129
Lung Cancer	21	22	24	24	27	28	28
Hayes-Roth	29	30	32	33	39	42	44
Car Evaluation	154	157	161	170	174	177	179
Lymphography	41	43	44	49	57	59	59
Spect Heart	40	41	42	45	53	54	54
Tic-Tac-Toe	106	109	110	115	123	124	125

These tables show that the belief rough set classifier based on generalization distribution table (BRSC-GDT) with their versions (H-BRSC-GDT, D-BRSC-GDT) give more combined decision rules for all databases and for all degrees of uncertainty than the belief rough set classifier (BRSC) with their versions (H-BRSC, D-BRSC). To understand that, let us remember that the BRSC-GDT which is based on the hyprid system GDT-RS selects only the not contradictory decision rules having best strength. For example, the mean size of the decision rules induced from *Balance Scale* database is 48 using BRSC-GDT and 59 using BRSC.

These tables also show that the belief rough set classifier based on generalization distribution table (BRSC-GDT) with their versions (H-BRSC-GDT, D-BRSC-GDT) give more combined decision rules for all databases and for all degrees of uncertainty than the pruned BDT. For example, the mean size of the decision rules induced from *Solar Flares* database is 101 using BRSC-GDT and 121 using pruned BDT.

From these tables, we can also see that the pruned BDT takes the middle place. The latter gives more combined belief decision rules for all databases and for all degrees of uncertainty than the standard belief rough set classifier (BRSC) with their versions (H-BRSC, D-BRSC). For example, the mean size of the belief decision rules induced from *W. Breast Cancer* database is 45 using pruned BDT and 50 using BRSC. This result is also explicable since the pruning step reduce the size of the overfitting decision tree.

From these tables, we found that our heuristic feature selection method and our dynamic reduct method used in a pre-processing stage of the BRSC and the BRSC-GDT increase slightly the size of the generated models for all databases. For example, the mean size of the decision rules induced from *Lung Cancer* database is 27 using BRSC and 28 using H-BRSC. The mean size of the decision rules induced from *Spect Heart* database is 40 using BRSC-GDT and 41 using H-BRSC-GDT.

Finally, we can also conclude that the size increases when the uncertainty increases. It is true for all databases and for all classifiers.

To illustrate the results, Figures 6.4, 6.5 and 6.6 give a graphical presentation for the mean size for all approaches relative to some databases namely $Nursery$, $Car\ Evaluation$ and $Tic - Tac - Toe$.

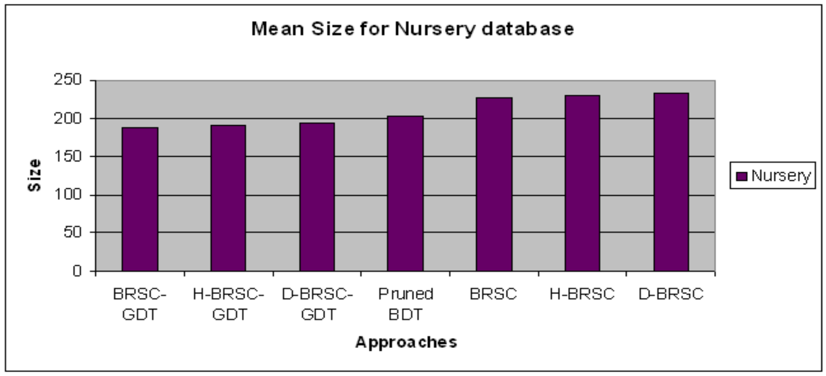

Figure 6.4: Mean size for Nursery database

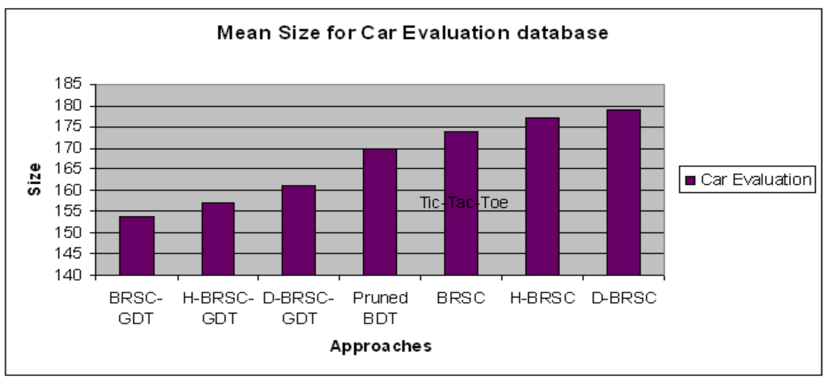

Figure 6.5: Mean size for Car Evaluation database

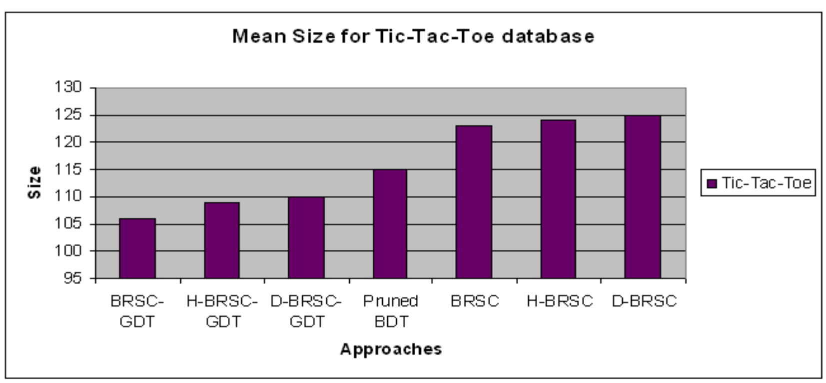

Figure 6.6: Mean size for Tic-Tac-Toe database

Classification accuracy criterion

Finally, we move to the most important evaluation criterion representing the classification accuracy of the learned decision rules generated from BDT, BRSC, BRSC-GDT, H-BRSC, H-BRSC-GDT, D-BRSC, D-BRSC-GDT. From Table 6.8 to Table 6.11, we detail the different results relative to the chosen databases for the certain and the uncertain cases where the different classifiers are sorted from the best to the worst. Table 6.12 summarizes the mean PCC for the different degrees of uncertainty from all databases relative to our classifiers.

Table 6.8: Experimental results for the PCC (%) (certain case)

Bases	D-BRSC	D-BRSC-GDT	BRSC	BRSC-GDT	H-BRSC	H-BRSC-GDT	Pruned BDT
W. Breast Cancer	90.51	87.62	86.51	85.03	84.87	83.61	83.53
Balance Scale	87.96	86.83	85.7	84.53	83.85	81.78	79
C. Voting records	98.94	98.94	98.71	98.06	97.98	97.57	98.53
Zoo	96.97	96.86	95.28	94.92	94.53	93.71	92.08
Nursery	98.92	98.78	97.89	97.66	96.84	96.79	96.07
Solar Flares	92.81	91.93	90.9	89.83	87.95	86.87	86.13
Lung Cancer	81.94	80.27	79.63	78.74	77.64	76.72	74.98
Hayes-Roth	98.97	98.73	98.70	97.74	95.82	94.78	84.04
Car Evaluation	85.83	85.94	84.11	83.72	82.95	81.77	73.62
Lymphography	86.56	86.16	85.78	84.98	84.78	84.57	79.59
Spect Heart	89.63	88.91	87.67	86.93	86.74	85.67	84.02
Tic-Tac-Toe	88.58	87.76	87.59	86.74	86.01	85.68	83.94

Table 6.9: Experimental results for the PCC (%) (low uncertainty)

Bases	D-BRSC	D-BRSC-GDT	BRSC	BRSC-GDT	H-BRSC	H-BRSC-GDT	Pruned BDT
W. Breast Cancer	89.87	86.98	86.41	84.74	84.41	83.77	83.46
Balance Scale	87.47	86.75	85.3	84.45	83.77	81.46	78.15
C. Voting records	98.94	98.76	98.69	97.98	97.58	97.44	98.28
Zoo	96.52	96.47	95.22	94.75	94.25	93.52	91.94
Nursery	98.68	98.49	97.44	97.43	96.34	96.06	95.84
Solar Flares	92.67	91.86	90.82	89.74	87.68	86.64	85.78
Lung Cancer	81.77	80.18	79.51	78.58	77.56	76.50	74.63
Hayes-Roth	98.96	89.62	98.53	97.65	95.65	94.46	83.66
Car Evaluation	85.46	84.71	83.91	83.44	82.79	81.46	73.49
Lymphography	86.35	86.14	85.22	84.76	84.75	84.24	79.25
Spect Heart	89.37	88.41	87.34	86.80	86.56	85.34	83.46
Tic-Tac-Toe	88.36	87.65	87.52	86.26	85.98	85.43	83.91

Table 6.10: Experimental results for the PCC (%)(middle uncertainty)

Bases	D-BRSC	D-BRSC-GDT	BRSC	BRSC-GDT	H-BRSC	H-BRSC-GDT	Pruned BDT
W. Breast Cancer	89.58	86.64	86.33	84.38	84.01	83.48	83.01
Balance Scale	87.32	86.29	85.23	84.16	83.56	80.21	77.83
C. Voting records	98.76	98.47	98.36	97.67	97.27	97.16	97.76
Zoo	96.47	96.24	95.16	94.38	94.13	93.47	91.36
Nursery	98.21	98.18	97.24	97.23	96.03	95.81	95.13
Solar Flares	92.61	91.75	90.73	89.53	87.64	86.61	85.61
Lung Cancer	81.50	80.11	79.43	78.49	77.25	76.50	74.36
Hayes-Roth	98.15	98.51	98.38	97.43	95.32	94.11	83.31
Car Evaluation	85.17	84.33	83.83	83.34	82.71	81.17	73.11
Lymphography	86.13	86.05	84.96	84.53	84.50	84.03	78.97
Spect Heart	89.29	88.12	87.23	86.52	86.33	85.28	83.01
Tic-Tac-Toe	88.24	87.36	87.27	86.21	85.85	85.23	83.75

Table 6.11: Experimental results for the PCC (%)(high uncertainty)

Bases	D-BRSC	D-BRSC-GDT	BRSC	BRSC-GDT	H-BRSC	H-BRSC-GDT	Pruned BDT
W. Breast Cancer	89.18	86.21	86.06	84.25	83.67	83.05	82.17
Balance Scale	87.13	86.12	84.92	84.02	83.47	80.03	77.76
C. Voting records	98.47	98.23	98.11	97.54	97.01	96.92	97.71
Zoo	95.87	95.68	95.03	94.17	94.01	92.87	91.41
Nursery	98.07	98.02	97.14	97.12	95.72	95.27	95.11
Solar Flares	92.56	91.59	90.54	89.42	87.66	86.44	85.46
Lung Cancer	81.33	80.03	79.16	78.28	77.35	76.33	74.07
Hayes-Roth	97.75	97.28	97.24	97.29	95.11	94.05	82.14
Car Evaluation	85.01	84.29	83.75	83.08	82.09	81.10	72.97
Lymphography	85.76	85.96	84.73	84.37	84.04	83.67	78.94
Spect Heart	89.14	88.07	87.06	86.45	86.12	85.07	82.17
Tic-Tac-Toe	88.19	87.22	87.21	86.18	85.26	85.06	83.42

Table 6.12: Experimental results for the PCC (%)(mean)

Bases	D-BRSC	D-BRSC-GDT	BRSC	BRSC-GDT	H-BRSC	H-BRSC-GDT	Pruned BDT
W. Breast Cancer	89.78	86.68	86.32	84.60	84.24	83.47	83.04
Balance Scale	87.47	86.49	85.28	84.29	83.66	80.54	78.18
C. Voting records	98.77	98.60	98.46	97.81	97.46	97.27	98.07
Zoo	96.45	96.31	95.17	94.55	94.23	93.39	91.69
Nursery	98.47	98.36	97.42	97.36	96.23	95.98	95.53
Solar Flares	92.66	91.78	90.75	89.63	87.73	86.64	85.74
Lung Cancer	81.63	80.14	79.43	78.52	77.45	76.51	74.51
Hayes-Roth	98.45	98.28	98.21	97.52	95.47	94.35	83.28
Car Evaluation	85.36	84.56	83.90	83.39	82.63	81.37	73.29
Lymphography	86.20	86.07	85.18	84.66	84.51	84.12	79.17
Spect Heart	89.35	88.37	87.32	86.67	86.43	85.34	83.16
Tic-Tac-Toe	88.34	87.49	87.39	86.34	85.77	85.35	83.75

From these tables, we find that D-BRSC and D-BRSC-GDT are the more accurate approaches. This positive result is due to the dynamic reduct method which produces more stable and accurate decision rules from the uncertain decision table. So, our dynamic reduct method used in a pre-processing stage to the BRSC and the BRSC-GDT increases the classification accuracy of the generated models. For example, the mean PCC of the decision rules induced from *Lung Cancer* database is 79.43% using BRSC and 81.63% using D-BRSC. The mean PCC of the decision rules induced from *Lung Cancer* database is 78.52% using BRSC-GDT and 80.14% using D-BRSC-GDT.

However, we also find that H-BRSC and H-BRSC-GDT are the worst classification approaches based on rough sets. This negative result is due to the heuristic feature selection method which try to give quick result and not an optimal solution. Hence, our heuristic feature selection method used in a pre-processing stage to the BRSC and the BRSC-GDT decreases the classification accuracy of the generated models. For example, the mean PCC of the decision rules induced from $Solar\ Flares$ database is 90.75% using BRSC and 87.73% using H-BRSC. The mean PCC of the decision rules induced from $Solar\ Flares$ database is 89.63% using BRSC-GDT and 86.64% using H-BRSC-GDT.

These tables show that the models generated from BRSC are more accurate than those obtained from BRSC-GDT for all databases and for all degrees of uncertainty. For example, the mean PCC of the decision rules induced from $C.\ Voting\ records$ database is 98.46% using BRSC and 97.81% using BRSC-GDT. It also true for the D-BRSC which is more accurate than the D-BRSC-GDT. For example, the mean PCC of the decision rules induced from Zoo database is 96.45% using D-BRSC and 96.31% using D-BRSC-GDT. The same thing for the H-BRSC which is more accurate than the H-BRSC-GDT. For example, the mean PCC of the decision rules induced from $Nursery$ database is 96.23% using H-BRSC and 95.98% using H-BRSC-GDT. The reason for these results is that the BRSC is based on an exhaustive recherche to obtain the best set of decision rules.

Besides, these tables also show that all our approaches based on rough sets are more accurate than the pruned BDT. This positive result is due to the concept of rough sets which try to produce a minimal and optimal set of decision rules without affecting the classification power. Finally, we can also conclude that the PCC slitly decreases when the uncertainty increases.

It is true for all databases and for all classifiers. For example the PCC for the *Car Evaluation* database is equal to 84.11% in certain case and equal to 83.75% in high uncertain case using BRSC.

To illustrate the obtained results, Figures 6.7, 6.8 and 6.9 give a graphical presentation for the mean PCC for all approaches relative to some databases namely *Balance Scale*, *Solar Flares* and $Hayes - Roth$.

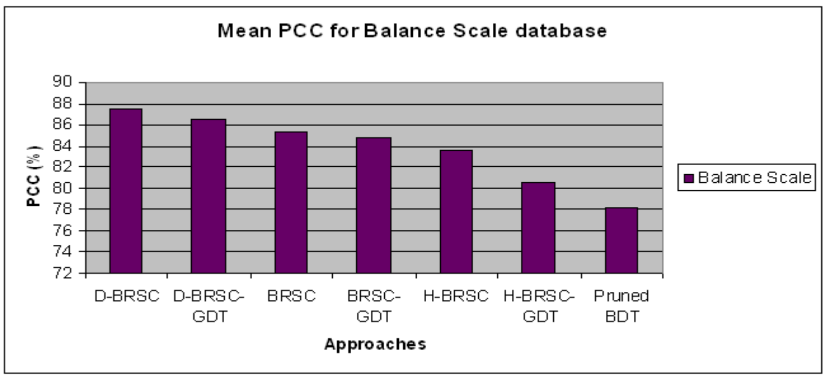

Figure 6.7: Mean PCC for Balance Scale database

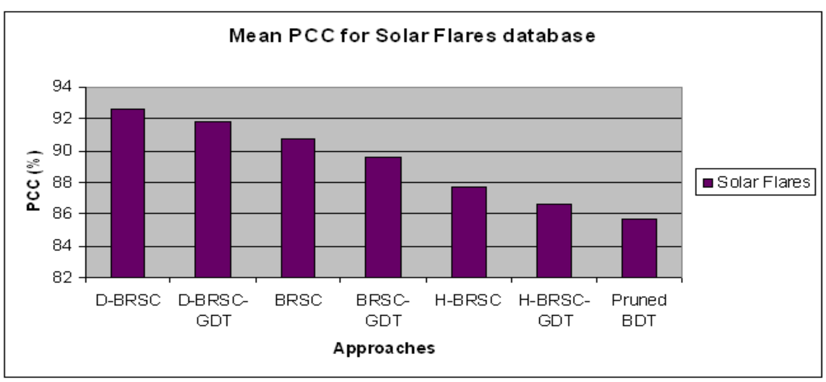

Figure 6.8: Mean PCC for Solar Flares database

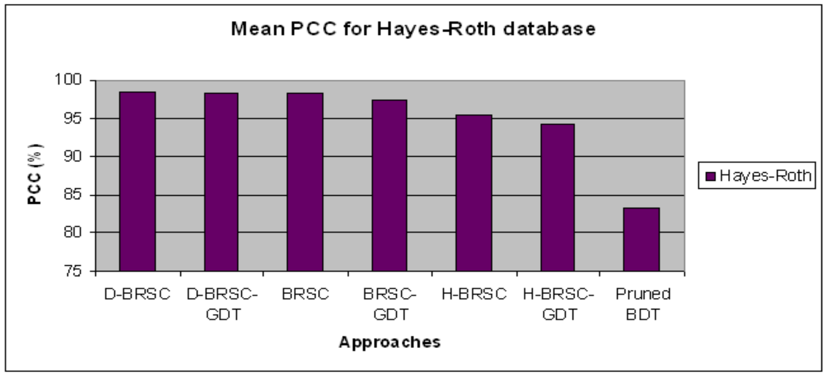

Figure 6.9: Mean PCC for Hayes-Roth database

6.3.2 Experimental results from real web usage database

In this subsection, we apply our classification approaches based on rough sets under belief functions (BRSC, BRSC-GDT, H-BRSC, H-BRSC-GDT, D-BRSC, D-BRSC-GDT) on our real web usage mining database. We compare the results with those obtained from the pruned BDT. Table 6.13 summarizes the results based on the three chosen evaluation criteria (time requirement, size of models and PCC) and relative to the certain case (using the crisp assignment of a visit to one of three clusters) and uncertain case (using the bba associated to each decision attribute).

Table 6.13: Experimental results relative to the web usage database

Approaches	Time requirement (seconds)	Size		PCC (%)	
		certain case	uncer case	certain case	uncer case
Pruned BDT	188	37	39	84.07	85.12
BRSC	157	41	46	85.16	89.63
BRSC-GDT	143	32	35	84.92	88.46
H-BRSC	127	43	47	84.91	87.84
H-BRSC-GDT	113	35	37	84.65	87.26
D-BRSC	332	43	47	86.24	91.63
D-BRSC-GDT	308	35	37	85.24	90.53

- For the *time requirement criterion*, we note that it is almost the same for the certain and the uncertain case. At first, we conclude that H-BRSC-GDT is the most fast approach relative to this web mining database. We also find that the classification approaches based on rough sets H-BRSC-GDT, H-BRSC, BRSC-GDT and BRSC are faster than the pruned BDT. The time requirement for the H-BRSC-GDT is equal to 113 seconds. However, the time requirement for the pruned BDT is 188 seconds. Figure 6.10 gives a graphical presentation for the time requirement needed for the building of each classification approach relative to the web usage database.

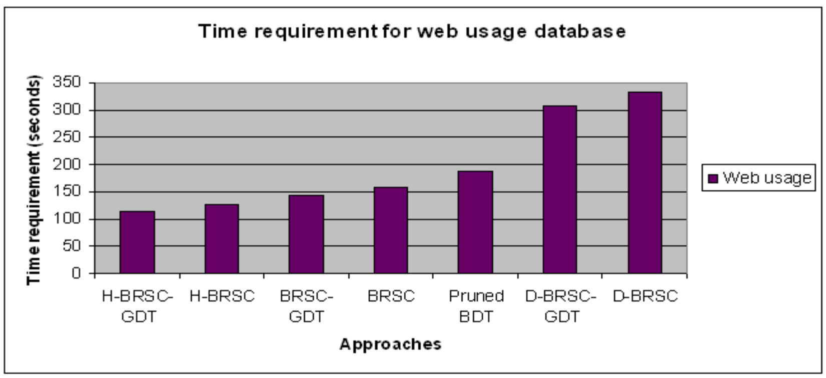

Figure 6.10: Time requirement for web usage database

- For the *size criterion*, we find that the size of the generated models relative to the all approaches based on the certain cluster assigns to each object is less than the size of the generated models based on the degree of beliefs associated to each cluster. For example, the size of the BRSC-GDT goes from 32 with certain case to 35 with uncertain case. We also conclude that the size of the model generated from the BRSC-GDT is the smallest one comparing with those obtained from the others. The number of the decision rules obtained from BRSC-GDT is equal to 35. Figure 6.11 gives a graphical presentation for the size of models for the all approaches relative to the web usage database according to the certain and the uncertain case.

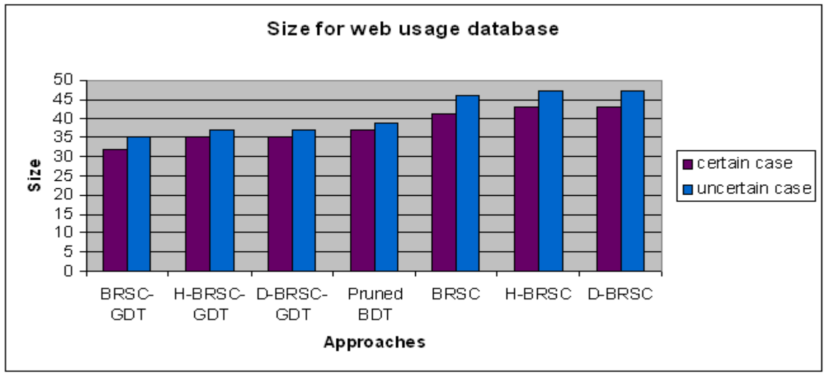

Figure 6.11: Size for web usage database

- For the *classification accuracy criterion*, we find that the PCC relative to the all approaches based on the certain cluster assigns to each object is less than the PCC based on the degree of beliefs associated to each cluster. For example, the PCC of the BRSC is equal to 85.24% with certain case. However, the PCC becomes 89.63% with uncertain case. We also conclude that the more accurate approach is D-BRSC. The PCC is equal to 91.63%. Figure 6.12 gives a graphical presentation for the classification accuracy criterion for the all approaches relative to the web usage database according to the certain and the uncertain case.

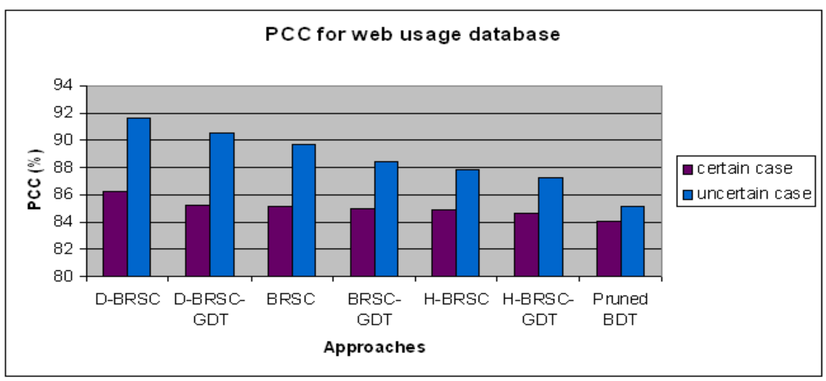

Figure 6.12: PCC for web usage database

From this experimental analysis, we deduce that the results relative to the web usage mining database are similar to those obtained from the U.C.I repository databases. This positive obtained results encourage interesting users to apply our classification approaches based on rough sets to web mining databases characterized by uncertain decision values by the means of belief functions.

6.3.3 Experimental conclusions

From the experimental results relative to our classification approaches detailed in the previous subsections, we can summarize by saying to the future user:

1. Use H-BRSC-GDT for quick model building which has the best time requirement for the construction procedure relative to the all databases.

2. Use BRSC-GDT for quick decision making which has the smallest model from the all classifiers relative to the all databases.

3. Use D-BRSC for more accurate decision making which has the best classification accuracy relative to the all databases.

However, we can advice the future user to use only the D-BRSC. First, because in the classification and the decision making problems the classification accuracy is in general the more important criterion. Second, the time requirement and the size of models relative to this classifier are not very bad.

6.4 Conclusion

In this chapter, we have performed experiments on real-world databases to judge the quality and the performance of our proposed classification systems based on rough sets according to three evaluation criterion: time requirement, size and classification accuracy. First, we have tested our belief classification approaches from modified U.C.I. repository databases where the uncertainty in decision attribute is created artificially using four degrees of uncertainty (no uncertainty, low uncertainty, middle uncertainty and high uncertainty). Then, we have tested them from a naturally uncertain web usage database. We have compared the results with those obtained from a similar classifier, denoted belief decision tree after the step of post-pruning.

We obtain interesting and positive results for all chosen databases and for all degrees of uncertainty regarding our new approaches based on rough sets comparing with those given by the pruned belief decision tree.

Conclusion

We have proposed in this thesis two new classification approaches based on rough sets that try to produce as possible a minimal and an optimal set of decision rules from an uncertain training decision table to classify new objects. We handle only symbolic condition and decision attribute values. The uncertainty appears only in decision attribute values and is represented by the *Transferable Belief Model* (TBM), one interpretation of the belief function theory.

In the first step of our thesis, we have generalized the basic concepts of rough sets under the belief function framework such as uncertain decision table, tolerance relation, set approximation, positive region, dependency of attributes, reduct and core. Besides, we have redefined the hybrid induction system GDT-RS for discovering classification decision rules in the new context to be called belief GDT-RS. These new definitions of the basic concepts of rough sets and the hybrid system, which are originally proposed in (Trabelsi & Elouedi, 2008, 2009), are useful to build our new classification systems based on rough sets.

The first proposed classification technique based on rough sets under belief functions is called *Belief Rough Set Classifier* (BRSC) (Trabelsi et al., 2009a) which is based on the new definition of the basic concepts of rough sets under belief function framework. The second classification technique is more sophisticated and is called *Belief Rough Set Classifier* based on *Generalization Distribution Table* (BRSC-GDT) (Trabelsi et al., 2010c) which is derived from the hybrid system named belief GDT-RS. The latter is a combination of *Generalization Distribution Table* (GDT) and *Rough Sets* (RS) under the belief function framework. Our classification techniques aim at generating a minimal and a significant set of decision rules that are suitable to classify new objects.

For both the classification techniques, we have detailed the same main phases relative to their building procedure such as the creation of the uncertain training decision table, the simplification of this uncertain training decision table and the generation of the decision rules. Note that the simplification of the decision table is the most important phase in the construction procedure relative to both the classifiers. This phase is different from each classification technique with different steps. Moreover, we have provided the classification procedure for the two approaches which is the same as the one used with standard rough set classifier except that we provide belief predictions rather than exact predictions.

Next, we have also proposed in this thesis two ideas to improve the construction procedure relative to the BRSC and the BRSC-GDT. The objective of the first idea is to improve the time requirement needed to build our models by applying in a pre-processing stage a heuristic feature selection method based on rough sets under the uncertain context (Trabelsi & Elouedi, 2010). This heuristic method can produce a sufficient solution without costly calculation. By applying it in the pre-processing stage of the BRSC and the BRSC-GDT, we obtain two other versions denoted by H-BRSC and H-BRSC-GDT.

On the other hand, to improve the classification power of the decision rules generated from our two classifiers, we have proposed as a second idea the notion of dynamic reduct under the belief function framework which yields more stable and accurate results (Trabelsi et al., 2009b, 2010b, 2011). By applying it in the pre-processing stage of the BRSC and the BRSC-GDT, we obtain two other versions denoted by D-BRSC and D-BRSC-GDT.

To evaluate the performance of our two classification techniques based on rough sets under uncertainty and their versions, we have implemented them with Matlab V6.5. Then, we have carried several experimentations on some U.C.I repository databases. The latters are artificially contaminated to include the uncertainty in the decision attribute values because of the lack of availability of real belief data sets. To further evaluate the quality of our two belief classification approaches and their versions, we have also performed experimentations (Trabelsi et al., 2010a) on a naturally uncertain web usage database where the uncertainty was not created artificially like U.C.I repository databases. This dataset was obtained from web access logs of the introductory computing science course at Saint Mary's University. Instead of using crisp assignment of a visit to one of the three clusters, a study patterns associated a basic belief assignment (bba). The resulting uncertain clustering was characterized using belief functions.

The more significant performance criteria used in the experimental phase are: the time requirement of learning, the size of models and the classification accuracy. Besides, we have compared the different results obtained from each classification system based on rough sets and also compared them with those given by a similar classifier, named the *Belief Decision Tree* (BDT). The latter is a decision tree built from uncertain dataset where the uncertainty appears also in decision attributes values and is represented through the TBM.

Experiments have shown the feasibility and the quality of the different proposed approaches. The results have shown that no approach has outperformed all others but each approach has shown its strengths in some particular situations. For the classification accuracy which is in general the first evaluation criterion considered for the most classification and decision making problems, we have found that the D-BRSC is the more accurate approach and the pruned BDT is the least one. Besides, we have also found interesting experimental results for the web usage mining database that may encourage users or experts in web domain to use our approaches to handle uncertainty in the decision attribute (Trabelsi et al., 2010a).

Some interesting future works have to be mentioned like *Distributed Data Mining* (DDM) which has evolved into an important and active area of research because of theoretical challenges and practical applications associated with the problem of extracting, interesting and previously unknown knowledge from very large real-world databases. Furthermore, some extensions of rough set theory (rough mereology) have brought new methods of decomposition of large data sets, data mining in distributed and multi-agent based environments (Polkowski & Skowron, 2001, 1999). Hence, we suggest as a future work, proposing new classification systems based on rough sets from uncertain databases to be applied for a lot of modern applications that fall into the category of systems that need DDM supporting distributed decision making. Real-world applications can be of different natures and from different scopes, for example, data and information fusion for situational awareness; scientific data mining in order to compose the results of diverse experiments and design a model of a phenomenon, intrusion detection, analysis, prognosis and handling of natural and man-caused disaster to prevent their catastrophic development, web mining, etc.

As a second future work, we also suggest handling another kind of uncertainty in the decision table like partially uncertain condition attribute values, in addition to uncertain decision attribute values. This uncertainty exists in real-world databases. For example, some condition or decision attribute values in a client's database, used by the bank to plan a loan policy, are missing or partially uncertain. This uncertainty can be also represented by belief function theory due to its advantages.

Finally, we also propose new classification approaches based on rough sets from decision tables characterized by uncertain quantitative condition or decision attributes. This uncertainty can be represented via the continuous belief functions. The objective is to create new classification systems based on rough sets from many kinds of databases.

Appendix A

Implementation

A.1 Introduction

In order to test our two new classification approaches, namely 'Belief rough set classifier' and 'Belief rough set classifier based on generalization distribution table', we have developed programs in Matlab V6.5. Obviously, we have implemented the basic concepts of rough sets under uncertainty, the building and the classification procedures of the two classifiers and the two ideas of improvement: the heuristic feature selection method and the dynamic reduct method.

As detailed in the previous chapters, these programs are developed to handle symbolic attributes. They have as input data sets with objects having certain condition attribute values but uncertain decision attribute values represented by basic belief assignments.

The output of our programs are basically:

1. A set of minimal decision rules induced from the two classifiers.

2. Results from the classification phase, especially the Percent of Correct Classification, denoted PCC.

This appendix is divided in two parts. The first one presents the major variables used in the different programs. The second part describes the different algorithms used to create our two classification approaches with their versions.

A.2 Major variables

In this section, we present the list of the major variables that are used in our developed algorithms:

- *training_set* includes the condition attribute values of all instances and the beliefs on their decision values.
- *number_training_objects* is the number of instances in the training set.
- *number_cond_attributes* is the number of condition attributes relative to the given classification problem.
- *number-decision-values* is the number of decision values relative to the given classification problem.
- *equivalence_classes* includes partitions of the training_set objects having the same condition attribute values.
- *tolerance_classes* includes partitions of the training_set objects having the same or similar decision attribute values.
- *threshold* is the threshold value needed to have flexible results.
- *lower_set* includes training_set objects having positive outcome.
- *upper_set* includes training_set objects having possible outcome.
- *positive_set* includes training_set objects having positive outcome for every decision attribute values.
- *k_degree* is the dependency degree of our training_set.
- *test_reduct* if a part of training_set is a reduct the variable will be true else it will be false.
- *core_set* includes the set of all indispensable attribute in the given classification problem.
- *reduct_value_set* includes the set of all indispensable attribute in the given classification problem.
- *decision_rules_BRSC* includes the set of the decision rules generated by the BRSC.

- *decision_rules_BRSC-GDT* includes the set of the decision rules generated by the BRSC-GDT.
- *size_model* is the number of the decision rules generated from the classifiers.
- *PCC* is the percent of correct classification relative to a testing set.
- *heuristic_reduct* includes one of the possible reduct without exhaustive research.
- *dynamic_reduct* includes the more stable reduct for the given classification problem.

A.3 Major programs

In this section, we present the different programs that we have developed in order to build and to test our classification approaches based on rough sets, namely the BRSC and the BRSC-GDT. The different programs are divided into three parts:

1. The programs relative to the basic concepts of rough sets under uncertainty.
2. The programs relative to the construction and the classification of BRSC and BRSC-GDT.
3. The programs relative to the two ideas of improvement of the BRSC and the BRSC-GDT, namely heuristic feature selection method and dynamic reduct method.

A.3.1 Programs relative to the basic concepts of rough sets under uncertainty

In this subsection, we present the different algorithms relative to the basic concepts of rough sets under uncertainty:

- **Indiscernibility_relation** computes the indiscernibility classes based on the condition attributes of the given classification problem.

  Algorithm: Indiscernibility_relation
  Input: training_set
  Output: equivalence_classes

```
begin
number_training_objects ← length(training_set(:,1))
number_attributes ← length(training_set(1,:))
number_cond_attributes ← number_attributes-1
for i ← 1 to number_objects do
object1 ← training_set(i,1:number_cond_attributes)
occ ← 0
    for j ← 1 to number_training_objects do
    object2 ← training_set(j,1:number_cond_attributes)
        if object1=object2
        then occ ← occ +1
        equivalence_classes(i,occ) ← j
        end if
    end for
end for
equivalence_classes ← Distinct(equivalence_classes)
end
```

- **Tolerance_relation** computes the tolerance classes based on the decision attribute of the uncertain decision table where 'Distance' is a function which computes the distance between two bba's.

Algorithm: Tolerance_relation
Input: training_set, number_decision_values, threshold
Output: tolerance_classes

```
begin
number_training_objects ← length(training_set(:,1))
number_attributes ← length(training_set(1,:))
size_power_set ← 2^{number−decision−values}
for i ← 1 to number_decision_values do
bba_certain ← zeros(1, size_power_set)
bba_certain(1,i+1) ← 1
occ ← 0
    for j ← 1 to number_training_objects do
    bba←training_set(j, number_attributes)
        if Distance(bba,bba_certain)< 1-threshold
        then occ ← occ +1
        tolerance_classes(i,occ) ← j
        end if
```

```
    end for
end for
end
```

- **Lower_approximation** computes the set of objects having positive outcome for a decision attribute value of the uncertain decision table. The function 'Average' combines two bba's using the mean operator.

  Algorithm: Lower_approximation
  Input: training_set, number_decision_values, threshold
  Output: lower_set

```
begin
number_training_objects ← length(training_set(:,1))
size_power_set=2^number_decision_values
equivalence_classes ← Indiscernibility_relation(training_set)
nb_equivalence_classes ← length(equivalence_classes(:,1))
tolerance_classes ← Tolerance_relation (training_set,
   number_decision_values, threshold)
for i ← 1 to number_decision_values do
bba_certain=zeros(1,size_power_set)
bba_certain(1,i+1)← 1
pos ← 1
    for j ← 1 to nb_equivalence_classes do
    equivalence_class ← equivalence_classes(j,:)
    size_equivalence_class ← length(equivalence_class)
    bba_equivalence_class ← Extraire_bba(base,equivalence_class)
    bba_combined ← Average(bba_equivalence_class)
        if Inclu (equivalence_class, tolerance_class) = true and
           Distance(bba_certain, bba_combined)< threshold
        then lower-set (i, pos:pos + size_equivalence_class - 1)←
               equivalence_class)
              pos ← pos + size_equivalence_class
        end if
    end for
end for
end
```

- **Upper_approximation** computes the set of objects having possible outcome for each decision attribute value of the given classification problem.

Algorithm: Upper_approximation
Input: training_set, number_decision_values, threshold
Output: upper_set
1. **begin**
2. number_training_objects ← length(training_set(:,1))
3. size_power_set=$2^{number\_decision\_values}$
4. equivalence_classes ← Indiscernibility_relation(training_set)
5. nb_equivalence_classes ← length(equivalence_classes(:,1))
6. tolerance_classes ← Tolerance_relation (training_set, number_decision_values, threshold)
7. **for** i ← 1 **to** number_decision_values **do**
8. pos ← 1
9. **for** j ← 1 **to** nb_equivalence_classes **do**
10. equivalence-class ← equivalence-classes(j,:)
11. size-equivalence-class ← length(equivalence_class)
12. **if** Intersection(equivalence_class, tolerance_class) = true
13. **then** upper_set (i, pos:pos+ size_equivalence_class - 1)← equivalence_class)
14. pos ← pos + size_equivalence_class
15. **end if**
16. **end for**
17. **end for**
18. **end**

- **Positive_region** computes the set of objects having positive outcome for all decision attribute value of the given classification problem.

Algorithm: Positive_region
Input: training_set, number_decision_values, threshold
Output: positive_set
1. **begin**
2. positive_set ← ∅
3. lower_set ← Lower_approximation(training_set, number_decision_values, threshold)
4. **for** i ← 1 **to** number_decision_values **do**
5. positive_set ← Concat (positive_set ,lower_set(i,:))
6. **end for**
7. **end**

- **Dependency_degree** computes the degree of dependency between condition and decision attributes of the uncertain decision table.

Algorithm: Dependency_degree
Input: training_set, number_decision_values, threshold
Output: k_degree
1. **begin**
2. number_training_objects ← length(training_set(:,1))
3. size_positive_set ← length(positive_region(training_set, number_decision_values, threshold))
4. k_degree ← size_positive_set/number_trainingobjects
5. **end**

- **Test_reduct** checks that a part of the training set is a reduct of the given classification problem.

Algorithm: Test_reduct
Input: training_set, part_training_set, number_decision_values, threshold
Output: test_reduct
1. **begin**
2. **if** Dependency_degree(training_set, number_decision_values, threshold) = Dependency_degree (part_training-set, number_decision_values, threshold)
3. **then** test_reduct← true
4. **else** test_reduct← false
5. **end if**
6. **end**

- **Relative_core** computes the indispensable condition attributes of the given classification problem.

Algorithm: Relative_core
Input: training_set, number_decision-values, threshold
Output: core_set
1. **begin**
2. number_training_objects ← length(training_set(:,1))
3. number_attributes ← length(training_set(1,:))
4. number_cond_attributes ← number_attributes - 1
5. part_training_set← zeros(number_training_objects, number_cond_attribute, number_cond_attribute)
6. part_training_set(:,1:number_cond_attributes-1,1) ← training_set(:, 2:number_cond_attributes-1)

7. part_training_set(:,number_cond_attributes, 1) ← training_set(:, number_attributes)
8. **for** i ← 2 **to** number_cond_attributes **do**
9. part_training_set(:,1 :i-1,i) ← training_set(:,1:i-1)
10. part_training_set(:,i :number_cond_attribute-1,i) ← training_set(:,i+1:number_cond_attribute)
11. part_training_set(:,number_cond_attribute,i)← training_set(:,number_attribute)
12. **end for**
13. **for** i ← 1 **to** number_cond_attribute **do**
14.   **if** Dependency_degree(training_set,number_decision-values, threshold) <> Dependency_degree (part_training_set(:,:,i), number_decision-values, threshold)
15.   **then** core_set ← concat(core_set, training_set(:,i))
16.   **end if**
17. **end for**
18. **end**

- **Reduct_value** computes the reduct values for the object j in the training set of the given classification problem.

Algorithm: Reduct_value
Input: training_set, j, number_decision_values, threshold
Output: reduct_value_set
1. **begin**
2. number_training_objects ← length(training_set(:,1))
3. number_attributes ← length(training_set(1,:))
4. number_cond_attributes ← length(training_set(1,:))-1
5. object_j ← training_set(j,:)
6. bba_object_j ← object_j(number_attributes)
7. **for** i ← 1 **to** number_cond_attributes **do**
8.   **for** k ← 1 **to** number_training_objects **do**
9.   bba_object_k ← training_set(k,number_attributes)
10.   **if** reducts_value_set (i)= training_set(k,i) and Distance(bba_object_k, bba_object_j) > threshold
11.   **then** object_j(i) ← 'vide'
12.   **end if**
13.   **end for**
14. **end for**
15. reducts_value_set ← object_j

16. **end**

A.3.2 Programs relative to the construction and the classification of BRSC and BRSC-GDT

In this subsection, we present the different major programs for the building and the classification phases relative to the BRSC and the BRSC-GDT:

- **Elimination_superfluous_attributes** eliminates the superfluous attributes for the training set of the given classification problem by computing the reduct.

  Algorithm: Elimination_superfluous_attributes
  Input: training_set, number_decision_values, threshold
  Output: reduct
  1. **begin**
  2. number_training_objects ← length(training_set(:,1))
  3. number_attributes ← length(training_set(1,:))
  4. reduct ← base
  5. parts_training_set ← zeros (number_training_objects, number_attributes-1, number_attributes - 1)
  6. parts_training_set (:,1:number_attributes - 2,1) ← base(:,2:number_attributes - 1)
  7. parts_training_set(:,number_attributes-1,1) ← base(:,number_attributes)
  8. **for** i ← 2 **to** number_attributes -1 **do**
  9. parts_training_set(:,1 :i-1,i) ← base(:,1:i-1);
  10. parts_training_set(:,i:number_attributes-2,i) ← base(:,i+1:number_attributes - 1)
  11. parts_training_set(:,number_attributes - 1,i) ← base(:,number_attributes)
  12. **end for**
  13. **for** i ← 1 **to** number_attributes - 1 **do**
  14. **if** Test_reduct(base,parts_training_set(:,:,i)) = true
  15. **then** Elimination_superfluous_attributes (parts_training_set(:,:,i), number_decision_values, threshold);
  16. **end if**
  17. **end for**
  18. **end**

- **Elimination_redundant_objects** eliminates the duplicate rows by combined them.

Algorithm: Elimination_redundant_objects
Input: training_set, number_decision_values, threshold
Output: training_set
1. **begin**
2. number_training_objects ← length(training_set(:,1))
3. number_attributes ← length(training_set(1,:))
4. **for** i ← 1 **to** number_training_objects **do**
5. object_i ← training_set (i,:)
6. bba_i ← training_set (i,number_attributes)
7.   **for** j ← 1 **to** number_training_objects **do**
8.   object_j ← training_set (j,:)
9.   bba_i ← training_set (j,number_attributes)
10.   **if** object_i = object_j and i <> j
11.   **then** training_set(j,condi training) ← Delete (training_set, i)
12.   **end if**
13.   **end for**
14.**end for**
15. **end**

- **Elimination_superfluous_attribute_values** eliminates some superfluous condition attribute values from our uncertain training set.

Algorithm: Elimination_superfluous_attribute_values
Input: training_set, position, number_decision_values, threshold)
Output: training_set
1. **begin**
2. number_training_objects ← length(training_set(:,1))
3. number_attributes ← length(training_set(1,:))
4. reduct ← base
8. **for** i← 1 **to** number_training_objects **do**
9. object ← Value_reduct (training_set, i, number_decision_values, threshold)
10. training_set (i,:) ← object
11.**end for**
12. **end**

- **BRSC_construction_procedure** generates using BRSC a set of significant decision rules from our uncertain training set.

Algorithm: BRSC_construction_procedure

Input: training_set, number_decision_values, threshold
Output: decision_rules_BRSC
1. **begin**
2. tic
3. reduct ← Elimination_superfluous_attributes (training_set, number_decision_values, threshold)
4. simplified_reduct ← Elimination_redundant_objects (training_set, number_decision_values, threshold)
5. value_reduct ← Elimination_superfluous_attribute_values (training_set, number_decision_values, threshold)
6. decision_rules_BRSC ← Elimination_redundant_objects (Value_reduct, number_decision_values, threshold)
7. toc
8. **end**

- **BRSC_classification_procedure** computes the PCC obtained from our set of decision_rules_BRSC relative the chosen testing_set.

  Algorithm: BRSC_classification_procedure
  Input: testing_set, decision_rules_BRSC, number_decision_values,threshold
  Output: PCC, size_model
  1. **begin**
  2. size_model ← length(decision_rules_BRSC (:,1))
  3. size_testing_set ← length(testing_set (:,1)
  4. number_attribute ← length(testing_set (1,:)
  5. **for** i ← 1 **to** size_testing_set **do**
  6. object ← testing_set(i,:)
  7. decision1 ← Max(pignistic (object(number_attribute)))
  8. bba_results ← Corresponding_rules (decision_rules_BRSC, object)
  9. bba_combined ← Averaging (bba_combined)
  10. decision2 ← Max(pignistic (bba_combined))
  11. **if** decision1=decision2
  12. **then** number_well_classified ← number_well_classified +1
  13. **end if**
  14. **end for**
  15. PCC ← (number_well_classified / size_testing_set)*100
  16. **end**

- **BRSC-GDT_construction_procedure** generates using BRSC a set of significant decision rules from our uncertain training set.

Algorithm: BRSC_GDT_construction_procedure
Input: training_set, number_decision_values, threshold
Output: decision_rules_BRSC-GDT
1. **begin**
2. tic
3. Compound_objects ← Elimination_redundant_objects (training_set, number_decision_values, threshold)
4. Compound_objects ← Delete_contradictory_objects (training_set, number_decision_values, threshold
5. size_Compound_objects ← length (Compound_objects)
6. **for** i ← **to** size_Compound_objects **do**
7. value_reduct_set ← Value_reduct (training_set,i, number_decision_values, threshold)
8. best_rule ← Max_strength(training_set, value_reduct_set)
9. decision_rules_BRSC-GDT (i,:) ← best_rule
10. **end for**
11. toc
12. **end**

- **BRSC_GDT_classification_procedure** computes the PCC obtained from our set of decision_rules_BRSC-GDT relative the chosen testing_set.

Algorithm: BRSC_GDT_classification_procedure
Input: training_set, decision_rules_BRSC-GDT,
number_decision_values, threshold
Output: PCC, size_model
1. **begin**
2. size_model ← length(decision_rules_BRSC-GDT (:,1))
3. size_testing_set ← length (testing_set (:,1)
4. number_attribute ← length (testing_set (1,:)
5. **for** i ← 1 **to** size_testing_set **do**
6. object ← testing_set (i,:)
7. decision1 ← object (number_attribute)
8. decision2 ← Corresponding_rules_BRSC-GDT (decision_rules_BRSC-GDT, object)
9. **if** decision1=decision2
10. **then** number_well_classified ← number_well_classified +1
11. **end if**
12. **end for**

13. PCC ← (number_well_classified/size_testing_set)*100
14. **end**

A.3.3 Programs relative to the two ideas of improvement of the BRSC and the BRSC-GDT

In this subsection, we give the algorithms relative to the feature selection method and to the dynamic reduct method which are developed in Chapter 5.

- **Heuristic_feature_selection_method** computes the reduct from our uncertain training set without exhaustive research and costly calculation.

  Algorithm: Heuristic_feature_selection_method
  Input: training_set, number_decision_values, threshold, heuristic_threshold
  Output: heuristic_reduct
  1. **begin**
  2. heuristic_reduct ← Relative_core (training_set, number_decision_values, threshold)
  3. remaining_training_set ← Delete_part_training_set(training_set, heuristic_reduct)
  4. **while** (Dependency_degree (heuristic_reduct, number_decision_values, threshold) < heuristic_threshold) **do**
  5. positive_set ← Positive_region (heuristic_reduct, number_decision_values, threshold)
  6. heuristic_reduct ← Delete_consistent_object (heuristic_reduct, positive_set)
  7. number_cond_remaining_att ← length (remaining_training_set(1,:))-1
  8. **for** i ← 1 **to** number_cond_remaining_att **do**
  9. heuristic_reduct1 ← heuristic_reduct
  10. value_v ← length(Positive_region(heuristic_reduct, number_decision_values, threshold))
  11. value_m ← length (Positive_region(heuristic_reduct, number_decision_values, threshold))
  12. value_x ← length (Positive_region(heuristic_reduct, number_decision_values, threshold))
  13. heuristic_cond ← value_v *value_m * value_x
  14. **end for**

15. **end while**
16. **end**

- **Dynamic_reduct_method** computes the more stable reduct from our uncertain training set. This algorithm need some subprograms such as 'Sampling' which can produce a part of our training set according to our sampling strategy (see subsection 6.2.4) and the 'More_ appearing_reduct' subprogram computes the occurrence of each reduct and gives the best.

  Algorithm: Dynamic_reduct_method
  Input: training_set, number_decision_values, threshold
  Output: dynamic_reduct
  1. **begin**
  2. **for** i ← 1 **to** 5 **do**
  3. part_training_set ← Sampling (training_set, i)
  4. reduct ← Elimination_superfluous_attributes (training_set, number_decision_values, threshold)
  5. possible_reduct ← Concat (possible_reduct, reduct)
  6. **end for**
  7. dynamic_reduct ← More_appearing_reduct (possible_reduct)
  8. **end**

A.4 Conclusion

In this appendix, we have presented the list of the major variables that are used in our developed programs and we have detailed the main algorithms relative to our new classification approaches based on rough sets from uncertain training set. These implementations are necessary to check the feasibility and the qualities of our solutions.

Appendix B

Web usage database

B.1 Introduction

This appendix gives a part (containing 210 objects) from the real web usage database (containing 7965 objects) described in the Chapter 6. Let us remember that the latter was obtained from web access logs of the introductory computing science course at Saint Mary's University. The course is 'Introduction to Computing Science and Programming' offered in the first term of the first year.

B.2 Web usage database structure

The table C.1 represents a part of the real web usage database. The different columns are described as follows:

- On/Off: to mean On campus/Off campus access.
- Day/Night: to mean Day time/Night time access.
- Class day: to mean Access during lab/class days or non-lab/class days.
- Hits: represents the number of hits.
- Notes: represents the number of class-notes downloads.
- Cluster: the visits fall into one of the following three clusters: Studious, Crammers, and Workers.
- m(cluster) and m(theta): represent the bba on the cluster.

Table B.1: Web usage database

N	On/ Off	Day/ Night	Class day	Hits	Notes	Cluster	m (cluster)	m (theta)
1	0	1	0	2	1	1	0.71	0.29
2	1	1	0	14	1	1	0.84	0.16
3	1	0	0	2	1	1	0.71	0.29
4	1	1	1	40	6	0	0.9	0.1
5	0	0	1	45	1	0	0.58	0.42
6	1	0	0	28	2	1	0.49	0.51
7	0	1	0	1	1	1	0.7	0.3
8	0	1	0	9	1	1	0.82	0.18
9	0	1	0	39	5	0	0.8	0.2
10	0	1	1	1	1	1	0.7	0.3
11	0	0	0	1	1	1	0.7	0.3
12	0	1	0	1	1	1	0.7	0.3
13	0	0	0	41	1	0	0.56	0.44
14	0	1	0	1	1	1	0.7	0.3
15	0	0	0	31	1	1	0.45	0.55
16	0	1	0	1	1	1	0.7	0.3
17	0	1	1	1	1	1	0.7	0.3
18	0	0	0	1	1	1	0.7	0.3
19	0	1	0	1	1	1	0.7	0.3
20	0	0	0	22	1	1	0.65	0.35
21	0	0	0	2	2	1	0.72	0.28
22	0	1	0	1	1	1	0.7	0.3
23	0	0	0	61	3	0	0.56	0.44
24	0	0	0	22	1	1	0.65	0.35
25	0	0	0	1	1	1	0.7	0.3
26	0	1	1	1	1	1	0.7	0.3
27	0	0	0	16	1	1	0.81	0.19
28	1	0	0	14	1	1	0.84	0.16
29	1	1	1	15	4	1	0.77	0.23
30	1	1	1	42	1	0	0.57	0.43
31	1	1	1	17	1	1	0.78	0.22
32	0	1	1	60	1	0	0.54	0.46
33	1	1	1	29	1	1	0.49	0.51
34	1	1	1	21	1	1	0.68	0.32
35	1	1	1	18	2	1	0.79	0.21
36	1	1	1	23	3	1	0.61	0.39
37	1	1	1	50	5	0	0.72	0.28

N	On/ Off	Day/ Night	Class day	Hits	Notes	Cluster	m (cluster)	m (theta)
38	1	1	1	66	41	2	0.5	0.5
39	1	1	1	11	1	1	0.84	0.16
40	1	1	1	21	1	1	0.68	0.32
41	1	1	1	15	2	1	0.9	0.1
42	1	1	1	15	2	1	0.9	0.1
43	0	1	1	32	1	0	0.45	0.55
44	0	0	1	55	4	0	0.63	0.37
45	1	0	1	36	2	0	0.55	0.45
46	1	0	1	32	2	0	0.48	0.52
47	1	0	1	16	2	1	0.86	0.14
48	0	0	1	15	1	1	0.83	0.17
49	1	1	0	12	5	1	0.72	0.28
50	0	1	0	56	1	0	0.55	0.45
51	1	1	0	58	8	0	0.58	0.42
52	1	1	0	23	1	1	0.63	0.37
53	1	1	0	18	2	1	0.79	0.21
54	1	1	0	15	2	1	0.9	0.1
55	0	1	0	31	3	0	0.49	0.51
56	1	1	0	21	1	1	0.68	0.32
57	1	1	0	41	5	0	0.85	0.15
58	1	1	0	58	7	0	0.6	0.4
59	1	1	0	21	1	1	0.68	0.32
60	1	1	0	18	1	1	0.76	0.24
61	0	1	0	51	6	0	0.71	0.29
62	0	1	0	30	2	1	0.45	0.55
63	0	1	0	11	1	1	0.84	0.16
64	0	1	0	21	1	1	0.68	0.32
65	0	1	0	10	1	1	0.83	0.17
66	1	0	0	52	3	0	0.63	0.37
67	0	0	0	28	5	0	0.48	0.52
68	0	0	0	31	4	0	0.53	0.47
69	0	0	0	15	1	1	0.83	0.17
70	0	0	0	13	3	1	0.91	0.09
71	0	0	0	27	2	1	0.52	0.48
72	1	1	0	29	1	1	0.49	0.51
73	1	1	0	15	4	1	0.77	0.23
74	1	1	0	18	2	1	0.79	0.21

N	On/ Off	Day/ Night	Class day	Hits	Notes	Cluster	m (cluster)	m (theta)
75	0	1	0	19	3	1	0.74	0.26
76	1	1	0	60	4	0	0.58	0.42
77	1	1	0	67	6	0	0.52	0.48
78	0	0	0	46	3	0	0.68	0.32
79	0	0	0	34	3	0	0.56	0.44
80	0	1	0	29	3	0	0.45	0.55
81	0	1	0	7	2	1	0.81	0.19
82	1	1	0	12	3	1	0.91	0.09
83	1	1	0	18	1	1	0.76	0.24
84	1	0	0	65	14	2	0.44	0.56
85	1	0	0	83	8	0	0.42	0.58
86	0	0	0	20	3	1	0.7	0.3
87	0	0	0	18	1	1	0.76	0.24
88	0	0	0	9	1	1	0.82	0.18
89	0	1	0	47	6	0	0.81	0.19
90	0	1	0	9	2	1	0.86	0.14
91	0	1	0	37	3	0	0.62	0.38
92	1	1	0	16	1	1	0.81	0.19
93	1	1	0	11	2	1	0.92	0.08
94	1	1	0	76	8	0	0.45	0.55
95	1	1	0	12	1	1	0.85	0.15
96	0	0	0	13	3	1	0.91	0.09
97	0	0	0	26	3	1	0.52	0.48
98	0	0	0	50	2	0	0.61	0.39
99	0	0	0	15	1	1	0.83	0.17
100	0	0	0	13	4	1	0.8	0.2
101	0	0	0	22	4	1	0.6	0.4
102	1	0	0	51	7	0	0.7	0.3
103	0	0	1	14	1	1	0.84	0.16
104	1	1	1	10	2	1	0.89	0.11
105	1	1	1	73	11	0	0.42	0.58
106	1	1	1	22	5	1	0.55	0.45
107	1	1	1	26	2	1	0.54	0.46
108	1	1	1	50	4	0	0.69	0.31
109	1	1	1	36	7	0	0.73	0.27
110	1	1	1	29	2	1	0.47	0.53
111	1	1	1	18	4	1	0.71	0.29

N	On/ Off	Day/ Night	Class day	Hits	Notes	Cluster	m (cluster)	m (theta)
112	1	1	1	34	1	0	0.48	0.52
113	1	1	1	100	9	0	0.38	0.62
114	0	1	1	20	1	1	0.71	0.29
115	1	1	1	29	3	0	0.45	0.55
116	1	1	1	65	1	0	0.51	0.49
117	1	1	1	31	2	0	0.46	0.54
118	1	1	1	36	2	0	0.55	0.45
119	1	1	1	30	3	0	0.47	0.53
120	1	1	1	15	2	1	0.9	0.1
121	1	1	1	53	1	0	0.57	0.43
122	1	1	1	15	2	1	0.9	0.1
123	1	1	1	85	19	2	0.53	0.47
124	1	1	1	57	7	0	0.61	0.39
125	1	1	1	55	7	0	0.64	0.36
126	1	1	1	17	1	1	0.78	0.22
127	1	1	1	27	3	1	0.49	0.51
128	1	1	1	16	1	1	0.81	0.19
129	1	1	1	10	1	1	0.83	0.17
130	1	1	1	40	5	0	0.83	0.17
131	1	1	1	33	6	0	0.64	0.36
132	1	1	1	8	3	1	0.82	0.18
133	1	1	1	12	1	1	0.85	0.15
134	1	1	1	30	3	0	0.47	0.53
135	1	1	1	24	1	1	0.6	0.4
136	1	1	1	16	1	1	0.81	0.19
137	1	1	1	39	4	0	0.72	0.28
138	1	1	1	25	1	1	0.58	0.42
139	1	1	1	56	6	0	0.63	0.37
140	1	1	1	14	2	1	0.93	0.07
141	1	1	1	46	5	0	0.8	0.2
142	1	1	1	5	1	1	0.76	0.24
143	1	1	1	18	1	1	0.76	0.24
144	1	1	1	11	3	1	0.89	0.11
145	1	1	1	38	4	0	0.7	0.3
146	1	1	1	22	3	1	0.64	0.36
147	1	1	1	11	2	1	0.92	0.08
148	1	1	1	18	4	1	0.71	0.29

N	On/ Off	Day/ Night	Class day	Hits	Notes	Cluster	m (cluster)	m (theta)
149	1	1	1	36	5	0	0.71	0.29
150	0	1	1	35	5	0	0.68	0.32
151	0	1	1	29	3	0	0.45	0.55
152	1	1	1	17	2	1	0.82	0.18
153	1	1	1	21	3	1	0.67	0.33
154	1	1	1	9	1	1	0.82	0.18
156	1	1	1	75	2	0	0.48	0.52
167	1	1	1	34	1	0	0.48	0.52
158	1	1	1	36	2	0	0.55	0.45
159	1	1	1	45	2	0	0.62	0.38
160	1	1	1	15	1	1	0.83	0.17
161	1	1	1	59	2	0	0.56	0.44
162	1	1	1	28	1	1	0.51	0.49
163	1	1	1	25	2	1	0.57	0.43
164	1	1	1	26	1	1	0.55	0.45
165	1	1	1	20	1	1	0.71	0.29
166	1	1	1	29	3	0	0.45	0.55
167	1	1	1	36	3	0	0.6	0.4
168	1	1	1	39	1	0	0.54	0.46
169	1	1	1	17	2	1	0.82	0.18
170	1	1	1	19	2	1	0.75	0.25
171	0	1	0	19	3	1	0.74	0.26
172	1	1	0	60	4	0	0.58	0.42
173	1	1	0	67	6	0	0.52	0.48
174	0	0	0	46	3	0	0.68	0.32
175	0	0	0	34	3	0	0.56	0.44
176	0	1	0	29	3	0	0.45	0.55
177	0	1	0	7	2	1	0.81	0.19
178	1	1	0	12	3	1	0.91	0.09
179	1	1	0	18	1	1	0.76	0.24
180	1	0	0	65	14	2	0.44	0.56
181	1	1	0	12	3	1	0.91	0.09
182	1	1	0	18	1	1	0.76	0.24
183	1	0	0	65	14	2	0.44	0.56
184	1	1	1	25	2	1	0.57	0.43
185	1	1	1	26	1	1	0.55	0.45

N	On/ Off	Day/ Night	Class day	Hits	Notes	Cluster	m (cluster)	m (theta)
186	1	1	1	36	5	0	0.71	0.29
187	0	1	1	35	5	0	0.68	0.32
188	0	1	1	29	3	0	0.45	0.55
189	1	1	1	17	2	1	0.82	0.18
190	1	1	1	21	3	1	0.67	0.33
191	1	1	1	9	1	1	0.82	0.18
192	1	1	1	75	2	0	0.48	0.52
193	1	1	1	34	1	0	0.48	0.52
194	1	1	1	36	2	0	0.55	0.45
195	1	1	1	45	2	0	0.62	0.38
196	1	1	1	15	1	1	0.83	0.17
197	1	1	1	59	2	0	0.56	0.44
198	1	1	1	28	1	1	0.51	0.49
199	1	1	1	25	2	1	0.57	0.43
200	1	1	1	26	1	1	0.55	0.45
201	1	1	1	20	1	1	0.71	0.29
202	1	1	1	29	3	0	0.45	0.55
203	1	1	1	36	3	0	0.6	0.4
204	1	1	1	39	1	0	0.54	0.46
205	1	1	1	17	2	1	0.82	0.18
206	1	1	1	19	2	1	0.75	0.25
207	1	1	1	29	3	0	0.45	0.55
208	1	1	1	36	3	0	0.6	0.4
209	1	1	1	39	1	0	0.54	0.46
210	1	1	1	26	1	1	0.55	0.45

B.3 Conclusion

In this appendix, we have presented a part of the web usage database characterized by uncertain decision attribute. The uncertainty is handled by belief functions. This real web mining database is needed to more judge the performance of our approaches.

Bibliography

An, A., Chan, C., Shan, N., Cercone, N., & Ziarko, W. (1997). Applying knowledge Discovery to predict Water-supply Consumption. *IEEE Expert*, *12*(4), 72-78.

Barnett, J. A. (1991). Calculating Dempster-Shafer Plausibility. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(6), 599-602.

Bauer, M. (1997). Approximations algorithms and decision making in the Dempster-Shafer theory of evidence- an empirical study. *International Journal of Approximate reasoning*, *17*(2), 217-237.

Bazan, J. G., Skowron, A., & Synak, P. (1994). Dynamic reducts as a tool for extracting laws from decision tables. In Z. W. Ras & M. Zemankova (Eds.), *Proceedings of the 8th International Symposium on Methodologies for Intelligent Systems, ISMIS'94* (pp. 346–355). Charlotte, North Carolina, USA: Springer-Verlag.

Ben-Hariz, S., Elouedi, Z., & Mellouli, K. (2006). Clustering approach using belief function theory. In *Proceedings of the 12th International Conference on Artificial Intelligence : Methodology, Systems, and Applications, AIMSA'06* (pp. 162–171). Varna, Bulgaria: Springer-Verlag.

Chouchoulas, A., & Shen, Q. (2001). Rough set-aided keyword reduction for text categorization. *Applied Artificial Intelligence: An International Journal*, *15*(9), 843-873.

Dempster, A. P. (1967). Upper and lower probabilities induced by a multiple valued mapping. *Annals of Mathematical Statistics*, *38*, 325-339.

Dempster, A. P. (1968). A generalization of Bayesian inference. *Journal of the Royal Statistical Society, Series B*, *30*, 205-247.

Denœux, T. (1995). A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics*, *25*(5), 804-813.

Denœux, T. (1999). Reasoning with imprecise belief structures. *International Journal of Approximate Reasoning*, *20*, 79-111.

Denœux, T. (2000). A neural network classifier based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics - Part A*, *30*(2), 131-150.

Denœux, T., & Skarstein-Bjanger, M. (2000). Induction of decision trees for partially classified data. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics* (pp. 2923–2928). Nashville, USA: IEEE.

Dong, J., Zhong, N., & Ohsuga, S. (1999). Probabilistic rough induction: The GDT-RS methodology and algorithms. In Z. W. Ras & A. Skowron (Eds.), *Proceedings of the 11th International Symposium on Foundations of Intelligent Systems* (pp. 621–629). Nashville, USA: Springer-Verlag.

Elouedi, Z., Mellouli, K., & Smets, P. (2001). Belief decision trees: Theoretical foundations. *International Journal of Approximate Reasoning*, *28*, 91-124.

Elouedi, Z., Mellouli, K., & Smets, P. (2002). A pre-pruning method in belief decision trees. In *The ninth international conference on information processing and management of uncertainty in knowledge-based systems, IPMU'02* (pp. 579–586). Annecy, France.

Elouedi, Z., Mellouli, K., & Smets, P. (2004). Assessing sensor reliability for multisensor data fusion within the transferable belief model. *IEEE Transactions on System Man and Cybernetics, Part B*, *34*, 782-787.

Fixsen, D., & Mahler, R. P. S. (1997). The modified dempster-shafer approach to classification. *IEEE Transactions on Systems, Man and Cybernetics - Part A*, *27*(1), 96-104.

Grzymala-Busse, J. W. (1991). On the Unknown Attribute Values in Learning from Examples. In *Proceedings of the ISMIS'91, Sixth International Symposium on Methodologies for Intelligent Systems* (pp. 368–377). Charlotte, NC: Springer-Verlag.

Grzymala-Busse, J. W. (1992). LERS-A system for learning examples based on rough sets. In R. Slowinski (Ed.), *Intelligent Decision Support, Handbook of Applications and Advances of the Rough Set Theory* (pp. 3–18). Boston, London.

Grzymala-Busse, J. W. (2003). Rough set strategies to data with missing attribute values. In *Workshop on Foundations and New Directions of Data Mining, the Third International Conference on Data Mining* (pp. 56–63). Melbourne, FL, USA.

Grzymala-Busse, J. W. (2004). Characteristic relations for incomplete data: A generalization of the indiscernibiliy relation. In *Proceedings of the RSCTC'2004, Fourth International Conference on Rough Sets* *and Current Trends in Computing* (pp. 244–253). Uppsala, Sweden: Springer-Verlag.

Grzymala-Busse, J. W., & Wang, A. Y. (1997). Modified algoritms LEM1 and LEM2 for rule induction from data with missing attribute values. In *Proceedings of the Fifth International Workshop on Rough Sets and Soft Computing (RSSC'97) at the Third Joint Conference on Information Sciences (JCIS'97)* (pp. 69–72). Research Triangle Park, NC.

Hong, T. P., Tseng, L. H., & Chien, B. C. (2002). Learning Fuzzy Rules from Incomplete Quantitative Data by Rough Sets. In *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'02* (pp. 1438–1443). Honolulu, Hawaii.

Janez, F. (1996). *Fusion de sources d'information définies sur des référentiels non exhaustifs différents.* Unpublished doctoral dissertation, Université d'Angers, France.

Jensen, R., & Shen, Q. (2003). Finding Rough Set Reducts with Ant Colony Optimization. In *Proceedings of the 2003 uk Workshop on Computational Intelligence* (pp. 15–22).

Jousselme, A. L., Grenier, D., & Bossé, E. (2001). A new distance between two bodies of evidence. *Information Fusion*, *2*(2), 91-101.

Kohlas, J., & Monney, P. A. (1995). *A mathematical theory of hints: An approach to Dempster-Shafer theory of evidence.* Lecture Notes in Economics and Mathematical Systems 425, Springer-Verlag.

Kryszkiewicz, M. (1995). Rough set approach to incomplete information systems. In *Proceedings of International Workshop on Rough Sets and Soft Computing at Second Annual Joint Conference on Information Sciences (JCIS'95)* (pp. 194–197).

Kryszkiewicz, M. (1999). Rules in imcomplete information systems. *Information Sciences*, *113*, 271-292.

Lingras, P. (1995). Plausibilistic rule extraction from incomplete databases using non transitive rough set model. In *Proceedings of the CSC'95 workshop on rough sets Database Mining* (pp. 213–226).

Lingras, P., & West, C. (2004). Interval set clustering of web users rough k-means. *Journal of Intelligent Information Systems*, *23*(1), 5-16.

Masson, M., & Denœux, T. (2004). Clustering Interval-valued Data using Belief Functions. *Pattern Recognition Lettres*, *25*(2), 163-171.

Mellouli, K. (1987). *On the propagation of beliefs in networks using the Dempster-Shafer theory of evidence.* Unpublished doctoral dissertation, School of business, University of Kansas.

Modrzejewski, M. (1993). Feature selection using rough sets theory. In *Pro-*

ceedings of the European Conference on Machine Learning, ECML'93 (pp. 213–226).

Murphy, C. K. (2000). Combining belief functions when evidence conflicts. *Decision Support Systems*, *29*(1), 1-9.

Nguyen, H. S. (1997). *Discretization of real value attributes, Boolean Reasoning Approach.* Unpublished doctoral dissertation, Warsaw University.

Pawlak, Z. (1981). Information systems-theoretical foundations. *Information Systems*, *6*(3), 205-218.

Pawlak, Z. (1982). Rough sets. *Journal of Computer and Information Sciences*, *11*, 341-356.

Pawlak, Z. (1991). *Rough sets: Theoretical aspects of reasoning about data.* Dordrecht: Kluwer Academic Publishing.

Piasta, Z., & Lenarcik, A. (1996). Rule induction with probabilistic rough classifiers. *ICS Research Report, Warsaw University of Technology.*

Polkowski, L., & Skowron, A. (1999). Towards adaptive calculus of granules. In *Computing with words in Information Intelligent Systems Foundations* (pp. 201–228). Physica Verlag.

Polkowski, L., & Skowron, A. (2001). Rough mereological calculi of granules: A rough set approach to computation. *Computational Intelligence*, *17*(3), 472-492.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning.* Morgan Kaufmann, San Mateo, California.

Rauszer, C. (1991). Reducts in information systems. *Fundamenta Informaticae*, *15*(1), 1-12.

Shafer, G. (1976). *A mathematical theory of evidence.* Princeton Univ. Press. Princeton, NJ.

Shafer, G. (1986). The combination of evidence. *International Journal of Intelligent Systems*, *1*, 155-180.

Siegel, S., & Castellan, N. J. (1988). *Non-parametric statistics for the behavioral sciences.* McGraw-Hill, Second edition.

Skowron, A. (1995). Synthesis of adaptive decision systems from experimental data. In *Proceedings of the fifth Scandinavian Conference on AI (SCAI'95)* (pp. 220–238). Trondheim, Norway: IOS Press.

Skowron, A., & Rauszer, C. (1992). The discernibility matrices and functions in information systems. In *Intelligent decision support- Handbook of the applications and advances of the rough sets theory* (pp. 331–362). Kluwer Academic Publishers.

Smets, P. (1988). Belief functions. In (pp. 253–286).

Smets, P. (1990). The combination of evidence in the transferable belief model. *IEEE Pattern analysis and Machine Intelligence*, *12*(5), 447-458.

Smets, P. (1992). The transferable belief model for expert judgments and reliability problems. *Reliability Engineering and System Safety*, *38*, 59–66.

Smets, P. (1995). The canonical decomposition of a weighted belief. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1896–1901). Montreal, Canada: IJCAI'95, Morgan Kaufmann.

Smets, P. (1998a). The application of the transferable belief model to diagnostic problems. *International Journal of Intelligent Systems*, *13*, 127-157.

Smets, P. (1998b). The transferable belief model for quantified belief representation. In D. M. Gabbay & P. Smets (Eds.), *Handbook of defeasible reasoning and uncertainty management systems* (Vol. 1, pp. 267–301). Doordrecht, The Netherlands: Kluwer.

Smets, P. (2002). Decision making in a context where uncertainty is represented by belief functions. In R. P. Srivastava (Ed.), *Belief functions in business decisions* (p. 17-61). Physica-Verlag.

Smets, P., & Kennes, R. (1994). The transferable belief model. *Artificial Intelligence*, *66*, 191–234.

Smets, P., & Kruse, R. (1997). The transferable belief model for belief representation. In A. Motro & P. Smets (Eds.), *Uncertainty in Information Systems: From needs to solutions* (pp. 343–368). Boston, MA: Kluwer.

Stefanowski, J. (1998). On rough set based approaches to induction of decision rules. In *Rough Sets in Data Minind and Knowledge Discovery* (pp. 500–529). Physica Verlag.

Stefanowski, J. (2001). *Algorithms of Decision Rule Induction in Data Mining.* Unpublished doctoral dissertation, Poznan University of Technology Press, Poznan, Poland.

Stefanowski, J., & Tsoukias, A. (2001). Incomplete information tables and rough classification. *Computational Intelligence*, *17*(3), 545-566.

Tessem, B. (1993). Approximations for efficient computation in the thoery evidence. *Artificial Intelligence*, *61*(2), 315-329.

Trabelsi, S., & Elouedi, Z. (2008). Learning decision rules from uncertain data using rough sets. In *Proceedings of the 8th International FLINS Conference* (pp. 109–114). Madrid, Spain.

Trabelsi, S., & Elouedi, Z. (2009). Attribute selection from partially uncertain data using rough sets. In *Proceedings of the Thrid International*

Conference on Moleding Simulation and applied optimization, ICMSAO'09. Sharjah, UAE.

Trabelsi, S., & Elouedi, Z. (2010). Heuristic method for attribute selection from partially uncertain data using rough sets. *International Journal of General Systems*, *39*(3), 271-290.

Trabelsi, S., Elouedi, Z., & Lingras, P. (2009a). Belief rough set classifier. In *Canadian AI 2009* (pp. 257–261). Springer-Verlag.

Trabelsi, S., Elouedi, Z., & Lingras, P. (2009b). Dynamic reduct from partially uncertain data using rough sets. In *Proceedings of the Twelfth International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, RSFDGrC 2009* (pp. 160–167). New Delhi, India: Springer-Verlag.

Trabelsi, S., Elouedi, Z., & Lingras, P. (2010a). Belief rough set classification for web mining based on dynamic core. In *Proceedings of the 10th International Conference on Intelligent Systems Design and Applications, international workshop on web mining and information retrieval* (pp. 403–408). Cairo, Egypt: IEEE.

Trabelsi, S., Elouedi, Z., & Lingras, P. (2010b). A comparison of dynamic and static belief rough set classifier. In *The seventh international conference on rough sets and current trends in computing, RSCTC 2010)* (pp. 366–375). Warsaw, Poland: Springer-Verlag.

Trabelsi, S., Elouedi, Z., & Lingras, P. (2010c). Rule discovery process based on rough sets under the belief function framework. In *Proceedings of 13th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2010* (pp. 726–736). Dortmund, Germany: Springer-Verlag.

Trabelsi, S., Elouedi, Z., & Lingras, P. (2011). Classification with dynamic reducts and belief functions. *Journal Transactions on Rough Sets*, *66*(14), 202-233.

Trabelsi, S., Elouedi, Z., & Mellouli, K. (2006). Pruning method of belief decision tree. *World Academy of Science, Engineering and Technology, Special Journal Issue*, *21*, 100-105.

Trabelsi, S., Elouedi, Z., & Mellouli, K. (2007). Pruning belief decision tree methods in averaging and conjunctive approaches. *International Journal of Approximate reasoning*, *46*, 568-595.

Vannoorenberghe, P., & Denœux, T. (2002). Handling uncertain labels in multiclass problems using belief decision trees. In *IPMU'2002* (pp. 598–601). Springer-Verlag.

Walley, P. (1991). *Statistical reasoning with imprecise probabilities.* Chapman and Hall, London.

Wroblewski, J. (1995). Finding minimal reducts using genetic algorithm. In *Proceedings of the 2nd Annual Joint Conference on Information Sciences* (pp. 186–189). Wrigthsville, North Carolina, USA.

Zhong, N., Dong, J., & Ohsuga, S. (1998). Data mining: A probabilistic rough set approach. In S.-A. Polkowski, L. (Ed.), *Rough sets in knowledge discovery* (pp. 127–146). Physica Verlag.

Zhong, N., Dong, J., & Ohsuga, S. (2001). Using rough sets with heuristics for feature selection. *Journal of Intelligent Information Systems*, *16*(3), 199-214.

Zhong, N., & Ohsuga, S. (1996). Using generalization distribution tables as a hypotheses search space for generalization. In *Proceedings of the fourth international workshop on rough sets, fuzzy sets, and machine discovery* (pp. 396–403). RSFD-96.

Ziarko, W. (1998). KDD-R: rough sets-based data mining system. In *Rough sets knowledge discovery part ii studies fuzziness and soft computing* (pp. 598–601). Springer-Verlag.

Zouhal, L. M., & Denœux, T. (1998). An evidence theoretic k-nn rule with parameter optimisation. *IEEE Transactions on Systems, Man and Cybernetics - Part C*, *28*(2), 263-271.

Résumé

Cette thèse propose deux approches de classification basées sur la théorie des ensembles approximatifs qui permettent de produire un ensemble de règles de décision à partir d'un ensemble d'apprentissage incertain. L'incertitude existe dans les valeurs de l'attribut décision et elle est représentée via le *modèle des croyances transférables*, une interprétation de la théorie des fonctions de croyance. La première technique de classification est nommée *classifieur crédibiliste des ensembles approximatifs*. Elle est basée sur une nouvelle définition des concepts de base de la théorie des ensembles approximatifs basée sur la théorie des fonctions de croyance. La deuxième technique de classification est plus complexe et elle est nommée *classifieur crédibiliste des ensembles approximatifs* basé sur *le tableau de distribution généralisée*. Cette technique est dérivée à partir d'un système hybride. Ce dernier est une combinaison entre *le tableau de distribution généralisée* et *les ensembles approximatifs*. Nos solutions ont pour objectif de générer un ensemble minimal et significatif de règles de décision pour la classification des nouveaux objets. Nous avons aussi proposé deux idées pour améliorer la procédure de construction relative à chaque technique de classification. L'objectif de la première idée est de réduire le temps nécessaire pour construire nos modèles en appliquant dans un premier niveau une heuristique pour la sélection d'attribut basée sur la théorie des ensembles approximatifs dans un contexte incertain. Afin d'améliorer la qualité de classification des règles de décision générées par les deux classifieurs, nous avons proposé comme deuxième idée la notion de reduct dynamique dans un environnement incertain pour avoir des résultats plus stables. Finalement, pour évaluer la performance de nos techniques de classification, nous avons effectué des expérimentations sur des bases de données réelles mais modifiées pour introduire l'incertitude et sur une base web naturellement incertaine. Nos expérimentations sont basées sur trois critères d'évaluation: le temps de construction, la taille des modèles et l'exactitude de classification. Nous avons aussi comparé les résultats avec ceux obtenus par un classifieur similaire nommé *arbre de décision crédibiliste*.

Mots clés : classification, incertitude, théorie des ensembles approximatifs, théorie des fonctions de croyance.

Abstract

This thesis proposes two classification approaches based on rough sets that are able to produce a set of decision rules from uncertain training decision table. The uncertainty only appears in decision attribute values and is handled through the *Transferable Belief Model*, one interpretation of the belief function theory. The first classification technique is called *belief rough set classifier* which is based on the new definition of the basic concepts of rough sets under belief function framework. The second is more sophisticated and is called *belief rough set classifier* based on *generalization distribution table* which is derived from an hybrid system. The latter is a combination of *generalization distribution table* and *rough sets*. Our solutions aim at generating a minimal and a significant set of classification decision rules. Next, we have also proposed in this thesis two ideas to improve the construction procedure relative to each classification technique. The objective of the first idea is to improve the time requirement needed to build our models by applying, in a pre-processing stage, a heuristic feature selection method based on rough sets under the uncertain context. In order to improve the classification power of the decision rules generated from our two classifiers, we have proposed, as a second idea, the dynamic reduct method under the belief function framework which yields more stable results. Finally, to evaluate the performance our classification techniques, we have carried experimentations on modified real-world databases under our uncertain context and on a naturally uncertain web usage database. Three evaluation criteria are chosen: the time requirement of learning, the size of models and the classification accuracy. Then, we have compared the results with those given by a similar classifier named the *belief decision tree*.

Keywords: classification, uncertainty, rough set theory, belief function theory.