Université de Tunis
Université d'Artois

THESE DE DOCTORAT

en vue de l'obtention du titre de docteur en

Informatique de Gestion

Génie Informatique et Automatique

Classifier ensemble under the belief function framework

Asma Trabelsi

Soutenue le 03/10/2018, devant le jury composé de:

Anne-Laure JOUSSELME	Directrice de Recherche, OTAN	Rapporteur
Christophe MARSALA	Professeur des universités, Sorbonne université	Rapporteur
Nahla BEN AMOR	Professeur des universités, ISG de Tunis	Examinateur
Sébastien DESTERCKE	Chargé de Recherche CNRS, UTC	Examinateur
Olivier COLOT	Professeur des universités, Université de Lille	Invité
Zied ELOUEDI	Professeur des universités, ISG de Tunis	Directeur de thèse
Eric LEFÈVRE	Professeur des universités, Université d'Artois	Directeur de thèse

Laboratoires:
Le Laboratoire de Génie Informatique et d'Automatique de l'Artois (LGI2A)
Le Laboratoire de Recherche Opérationnelle, de Décision et de Contrôle de processus (LARODEC)

Contents

Introduction **1**

1 Ensemble classifiers **7**

1.1 Introduction 7

1.2 Classifier generation 9

1.2.1 Diversity measures 9

1.2.2 Ensuring diversity 12

1.3 Classifier combination 14

1.3.1 Combining class labels 14

1.3.2 Combining continuous outputs 17

1.4 Ensemble size 20

1.5 Conclusion 20

2 Ensemble classifiers within the belief function framework **21**

2.1 Introduction 22

2.2 Basic concepts of the belief function theory 22

2.2.1 Knowledge representation 23

2.2.2 Special bbas 25
2.2.3 Discounting 27
2.2.4 The dissimilarity between two pieces of evidence 27
2.2.5 Combining information sources 28
2.2.6 Decision process 32
2.3 Ensemble systems within the belief function theory 33
2.3.1 Mandler and Shurmann method 34
2.3.2 Xu et al.'s method 35
2.3.3 Rogova's method 36
2.3.4 Al-Ani's method 37
2.3.5 The calibration method 37
2.4 Conclusion 38

3 New evidential classifiers **39**

3.1 Introduction 39
3.2 Decision tree classifiers for evidential data 40
3.2.1 Standard decision trees 40
3.2.2 The structure of evidential training data 45
3.2.3 Decision tree parameters 46
3.2.4 Decision tree procedures 57
3.3 Enhanced Evidential *k*-Nearest Neighbors (EE*k*-NN) 65
3.4 Comparative study 68
3.4.1 Experimentation settings 68
3.4.2 Results discussions 70
3.5 Conclusion 74

4 A selective ensemble EEk-NN classifiers through rough set reducts 75

4.1 Introduction 75

4.2 Rough sets: Fundamental concepts 76

4.3 A rough set based selective ensemble EEk-NNs 80

4.3.1 Reduct extraction from data with evidential attributes . . . 80

4.3.2 Reduct selection for ensemble EEk-NN classifiers 82

4.4 Experimentation settings and results 88

4.4.1 Ensemble size 88

4.4.2 Reduct diversity 90

4.5 Conclusion 91

5 Combining selective ensemble EEk-NNs 93

5.1 Introduction 93

5.2 Dempster's rule for merging classifiers 94

5.3 Mixed rule for combining classifiers 100

5.3.1 Classifier diversity 100

5.3.2 The optimized t-norm for evidential classifier combination 101

5.3.3 A comparative study 103

5.4 Conclusion 105

Conclusion 107

Author publications 111

References 113

Acknowledgments

I would like to extend my grateful thanks to my supervisors Pr. Zied Elouedi and Pr. Eric lefèvre for their constructive discussions, their supports, their motivations and their guidance allowing the successful fulfillment of this Thesis.

I would like also to express my deepest gratitude to the thesis committee for their acceptance to judge this work and allowing us to make some improvement thanks to their interesting suggestions and constructive comments.

My thanks go to all members of the LARODEC and the LGI2A Laboratory, more especially Yoann Kubera and Nathalie Morganti.

I would like to thank Campus France for their financial support during the last year of my PhD.

And finally, I would like to express my love to my family for their unfailing support through these last three years. They were a source of encouragement and motivation to obtain the PhD diploma.

List of Figures

1 Theoretical aspects of the thesis 5
2 Thesis contributions 6

1.1 Ensemble classifier system 8
1.2 Decision profile matrix 18
1.3 Fusion operator example 18

2.1 The generic structure of the TBM framework 33
2.2 General framework of ensemble systems with belief function framework 34
2.3 Confusion Matrix 35

3.1 Decision tree according to the Gain Ratio Criterion 60
3.2 Decision tree according to the Diff Ratio Criterion 61
3.3 Results for Heart database 70
3.4 Results for Japanese database 71
3.5 Results for Vote Records database 71
3.6 Results for Hepatitis database 71
3.7 Results for Wine database 72

3.8 Results for Thoracic Surgery database 72

4.1 The illustration of the set approximation 77

4.2 A rough set based ensemble for data with evidential attributes . . 81

4.3 Reduct selection for ensemble learning 86

5.1 PCC results for *k* equals 7 97

5.2 Recall results for *k* equals 7 97

5.3 Precision results for *k* equals 7 98

5.4 The optimized t-norm based rule strategy 102

5.5 A comparison of the PCC results for a value of *k* equals 7 105

List of Tables

1.1 Distribution of decisions between two classifiers 10

1.2 Pairwise diversity measures 12

1.3 An example of the BKS table 16

1.4 Borda count example 17

3.1 An example of training data within a certain environment 42

3.2 Evidential training data 46

3.3 The probability of belonging of objects in terms of the attribute values. 50

3.4 Average Probability associated to the attribute Outlook 51

3.5 Average Probability associated to the attribute Temperature 51

3.6 Average Probability associated to the attribute Humidity 51

3.7 Average Probability associated to the attribute Windy 51

3.8 Probability distribution of leaves according to the *GainRatio* Criterion 59

3.9 Probability distribution of leaves according to the *DiffRatio* Criterion 59

3.10 Evidential attribute values for a query instance 63

3.11 Distance between query instances 67
3.12 Description of databases 69
3.13 Comparative results in terms of the PCC criterion (%) 73
3.14 Comparative results in terms of the recall criterion (%) 73
3.15 Comparative results in terms of the precision criterion (%) 74

4.1 An example of a discernibility matrix 79
4.2 Reduct generation for the different databases 89
4.3 Ensemble size for Heart database 89
4.4 Ensemble size for Japanese database 89
4.5 Ensemble size for Vote Records database 89
4.6 Ensemble size for Hepatitis database 89
4.7 Ensemble size for Thoracic Surgery database 90
4.8 Ensemble size for Wine database 90
4.9 Reduct diversity for Heart database 91
4.10 Reduct diversity for Japanese database 91
4.11 Reduct diversity for Vote Records database 91
4.12 Reduct diversity for Hepatitis database 91
4.13 Reduct diversity for Thoracic Surger database 91
4.14 Reduct diversity for Wine Data database 91

5.1 Combination results for Heart database 95
5.2 Combination results for Japanese database 95
5.3 Combination results for Vote Record database 96
5.4 Combination results for Hepatitis database 96
5.5 Combination results for Wine database 99

5.6 Combination results for Thoracic Surgery database 99

5.7 The degree of independence between individual classifiers 101

5.8 Combination results: PCC 104

List of Algorithms

4.1 Select diverse reducts (RD) 83
4.2 ReductDiversity(*RED_Chosen*,*RED*) 84
4.3 A reduct selection approach based on the Accuracy-Diversity assessment function 87

Introduction

Recent years have seen the revival of artificial intelligence (AI) and machine learning in particular, both in academic circles and the industry. What made these successes possible, in large part, is impetus to the need to process the huge amounts of data and the increase in raw computational power. Among the areas of AI is heavily used, machine learning has seen spectacular developments, and continues to find applicability in a wide range of domains. Its enormous success has spread its applications in different fields such as searches, speech recognition, and personal assistants on mobile phones, etc.

The ever-growing availability of data place an increasing demand on a new and more performant learning techniques. Therefore, in the last decade, we have seen the growth of many learning techniques from which we can cite several well known techniques such as ensemble learning (Quost, Masson, & Denœux, 2011), deep learning (LeCun, Bengio, & Hinton, 2015) and statistical learning (Vapnik, 1999) to cite a few.

Ensemble learning is one of those performing techniques that help improve machine learning results by combining several models. It relies on getting the best of several heterogeneous learning techniques using a fusion rules (Weiss & Kapouleas, 1989). Thus, this approach allows the production of better predictive/classification performance compared to a single model (Rhéaume & Jousselme, 2003; Quost et al., 2011). That is why ensemble methods placed first in many prestigious machine learning competitions, such as the Netflix Competition, KDD 2009, and Kaggle (Zhou, 2012).

The construction of an ensemble system and more specifically an ensemble classifier requires two main steps. The first one concerns the selection of diverse set of individual classifiers, while the second one consists of combining the output predictions of the selected classifiers. The choice of the individual classifiers as well as the combination operator could influence the ensemble performance. Ensuring diversity between the base classifiers has been defended as a successful means for the production of a performing ensemble system (Zhou, 2012). Diversity can be ensured in different manners. However, of all ensuring diversity method, diversifying of the input feature space has proven to be theoretically and experimentally among the most efficient and performing (Bryll, Gutierrez-Osuna, & Quek, 2003; Tumer & Ghosh, 1996; Turner & Oza, 1999). In fact, it does not only allow the minimization of the correlation between the combined classifiers, but it also performs faster thanks to the reduced size of the input feature space (Bryll et al., 2003; Günter & Bunke, 2004; Y. Kim, 2006; Tumer & Oza, 2003).

The process of generating feature subsets with good predicting potential is still undergoing research study. The Random Subspace Method (RSM) also called random subspacing is often used in the literature (Kotsiantis & Kanellopoulos, 2012). The major shortcoming of this latter technique is the random partition of the original input features. As a matter of fact, the random selection may potentially increase the risk of irrelevant and redundant features as part of the selected subsets.

Despite the promising results of RSM based ensemble classifier, several other techniques and frameworks have been used to enhance the performance of prediction. From the framework, we single out the mathematical theory of the rough set theory that has been successfully used to reduce the set of feature of any dataset (Bhattacharjee, Basu, Nasipuri, & Kundu, 2010). Introduced by Pawlak (Pawlak, 1982), the rough set theory has been successfully applied in pattern recognition, data mining and machine learning domains, more particularly for attribute reduction problems. The reduced attribute set, representing the minimal subset of attributes that enables the discrimination of objects with different decision values, is referred to as reduct. The concept of ensemble classifiers through rough set reducts have been introduced and applied in a wide range of practical problems such as text classification (Shi, Ma, Xi, Duan, & Zhao, 2011), biomedical classification (Shi, Xi, Ma, Weng, & Hu, 2011), tumor classification (Wang, Li, Zhang, Gui, & Huang, 2010), web services classification (Saha, Murthy, & Pal, 2008), etc. Nevertheless, in spite of their great importance, rough set ensemble classifiers, it has not been applied yet on imperfect data (TRABELSI & ELOUEDI, 2008).

In real world, the collected data suffer for noises, missing values that makes it imperfect and very complex to handle. This imperfection is due to multiple external factors such as obstacles, interference, missing information, etc. Uncertainty is generally represented through several uncertain framework such as probabilities, fuzzy set theory and evidential theory. The latter theory has been widely used in representing imperfect information in databases (Vannoorenberghe & Denœux, 2002). This data structure with evidential attribute is denoted as the evidential database. (Samet, Lefèvre, & Yahia, 2014). Despite its asset in modeling imperfect knowledge, this kind of database brought more complexity in its handling and remains time consuming. Therefore, using rough set ensemble classifiers on this database have never been more challenging. To unlock this research field, we propose, in this dissertation, a rough set based ensemble for processing data with evidential attributes.

With the lack of classification algorithms processing data with evidential attributes, we aim to construct novel evidential classifiers by extended some well-known existing classifiers. From the plethora of machine learning algorithms, we focus our interest on the decision tree and the *k*-Nearest Neighbors classifiers.

Decision tree constructs classification models in the form of a tree structure. It breaks down a training set into smaller subsets to develop the associated decision tree. The final output is a tree with decision and leaf nodes allowing the classification of new test patterns. The C4.5 proposed by Quinlan is the core algorithm for constructing decision trees (Quinlan, 1986). For the *K*-NN classifier, a distance between a test pattern and all the training patterns is computed. This distance can be computed either by the Euclidian or the Manhattan distances. The probable classes receive a vote from each of the *k* patterns that are closest to the test pattern in terms of the chosen distance. The class with the highest vote is considered to be the class of the test pattern.

Although the *k*-NN and the decision trees algorithms are widely used, they have not the ability to process data with evidential attributes. Thus, the first contribution of this thesis is to construct two decision tree versions and a *k*-NN classifier, called Enhanced Evidential *k*-NN (EE*k*-NN), for processing data with evidential attributes. A comparative study between the proposed classifiers is done to pick out the best one. Accordingly, the most performant algorithm is used as a base classifier for constructing rough set based ensemble. Our framework consists on two main steps: reduct generation and reduct selection for training individual classifiers.

As regards reduct generation process, almost all existing heuristics within the standard case are based on the computation of a discernibility matrix. An example includes the SAVGenetic reducer implemented within the Rosetta software (El-Monsef, Seddeek, & Medhat, 2003). This algorithm consists firstly of computing a discernibility matrix and the non empty set of the obtained matrix will then be used for picking out approximate hitting sets, meaning approximate reducts. Since it has yielded satisfactory results, we propose to extend the SAVGenetic algorithm in the context of evidential data for generating all possible reducts. Recognizing that hundreds or even thousands of reducts may be generated, the most suitable ones have to be used for constructing an ensemble classifier with good predict prower. Three reduct selection approaches have been investigated throughout this thesis. Suppose that *M* is the ensemble classifier size. Our first method, named Diversity Reduct (DR), consists of selecting *M* diverse reducts from the original pool of reducts. The second technique, called Accuracy Diversity Assessment Function (*AD-AF*), allows to select at most *M* reducts by taking into consideration not only the diversity between reducts but also the diversity and the accuracy between the individual classifiers. The third and the last approach is the Ensemble Accuracy Assessment Function (*EA-AF*) and it consists of selecting at most *M* diverse classifiers that maximize the ensemble accuracy. A comparative study between the proposed methods is made to examine the impact of the feature subspaces on classifier ensemble performance.

Concerning the classifier fusion process, the choice of the most appropriate combination rule is a crucial task to achieve performance. In the context of evidential classifier fusion, we relied mainly on the belief function combination rules. There exist several rules some of theme deal with the case of distinct sources, while others assume the independence between sources. Initially, we suppose the independence between classifiers that are trained from diverse feature subsets and we rely on a independent combination rule, notably the Dempster one. However, a study conducted by Quost et al. (2011) has proven the non efficiency of the Dempster rule when it comes to merge an ensemble of classifiers. An optimized t-norm based rule, with behavior ranging between the Dempster rule (i.e for independent classifier fusion) and the cautious rule (i.e for dependent classifier fusion), has already been suggested as an alternative. In this thesis, we aim to evaluate and compare the Dempster rule, the cautious rule and the t-norm optimized based rule in the context of rough set based classifier ensemble. The idea behind the comparative study is to identify the most appropriate fusion rule (i.e the rule achieving the high performance).

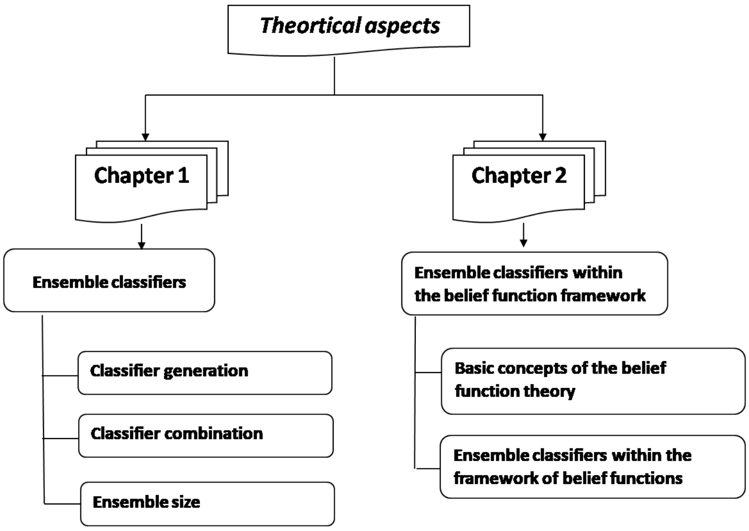

Theortical aspects
Chapter 1
Chapter 2
Ensemble classifiers
Classifier generation
Classifier combination
Ensemble size
Ensemble classifiers within the belief function framework
Basic concepts of the belief function theory
Ensemble classifiers within the framework of belief functions

Figure 1: Theoretical aspects of the thesis

This dissertation is structured into two substantial parts.

The first part, named Theoretical aspects, comprised two main Chapters (see Figure 1):

- Chapter 1: Ensemble classifiers. This chapter outlines the substantial factors enabling the construction of a good ensemble of classifier.
- Chapter 2: Ensemble classifier within the belief function framework. This Chapter is devoted to highlighting the basic concepts behind the belief function theory as well as the concern about ensemble classifier within the framework of belief functions.

The second part of this dissertation presents our main contributions (see Figure 2). It consists of three main chapters as follows:

- Chapter 3: Evidential classifiers. In this Chapter, we propose three machine learning classifiers for handling data with evidential attributes.
- Chapter 4: A selective ensemble EE*k*-NN classifiers through rough set reducts. We propose, in this Chapter, a novel framework for selecting individual classifiers allowing the highest ensemble performance.
- Chapter 5: Combining selective ensemble EE*k*-NN classifiers. This Chapter addresses the combination process. Concretely, we carry out a comparative

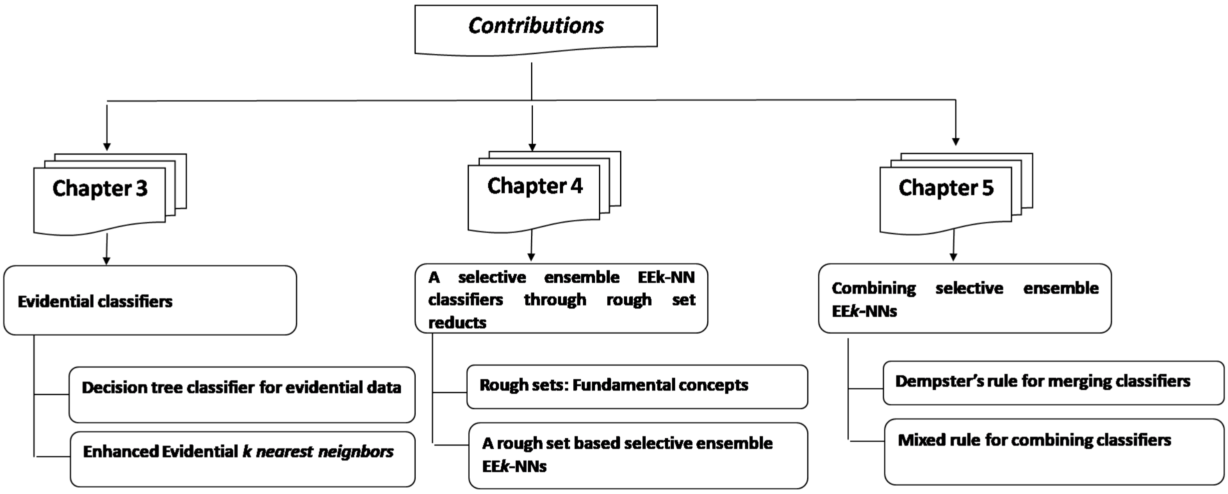

Contributions
Chapter 3
Chapter 4
Chapter 5
Evidential classifiers
Decision tree classifier for evidential data
Enhanced Evidential k nearest neignbors
A selective ensemble EEk-NN classifiers through rough set reducts
Rough sets: Fundamental concepts
A rough set based selective ensemble EEk-NNs
Combining selective ensemble EEk-NNs
Dempster's rule for merging classifiers
Mixed rule for combining classifiers

Figure 2: Thesis contributions

study between some well-known combination rules, namely the Dempster rule, the cautious rule and the optimized t-norm based rule. Our aim is to select the most suitable operator for an ensemble of EEk-NN classifiers.

Finally, a general conclusion sumps up the main contribution of this Thesis and presents some possible future work directions.

Chapter 1

Ensemble classifiers

Contents

1.1 Introduction 7
1.2 Classifier generation 9
  1.2.1 Diversity measures 9
  1.2.2 Ensuring diversity 12
1.3 Classifier combination 14
  1.3.1 Combining class labels 14
  1.3.2 Combining continuous outputs 17
1.4 Ensemble size 20
1.5 Conclusion 20

1.1 Introduction

Ensemble classifiers, also refereed to as ensemble methods, have been theoretically and experimentally (Hansen & Salamon, 1990) proven to be powerful techniques for improving the prediction performance of pattern recognition problems since the 1990s. The idea behind ensemble methods is to construct a predictive model by merging multiple ones. So that, various machine learning algorithms may potentially offer complementary information about query patterns, which could improve the performance of the individual classifiers (Quost et al., 2011).

The success of ensemble methods has been attributed to various reasons, notably statistical, computational and representational ones (Dietterich, 2000). There are also preferred when dealing with some application problems, especially those with a huge amount of data. A commonly used solution is to partitioning the original data into smallest subsets and then learning a classification algorithm with each subset of data. The prediction yielded by all classifiers are then merged through to get an accurate final decision. Ensemble classifiers are also well suited in the case of online learning. Since classifiers trained with the original data cannot be updated to learn new inputs, it is preferable to learn a novel individual classifier with the newest information sources and the obtained results will be merged with the earlier ones. The general structure of an ensemble classifiers is depicted in Figure 1.1.

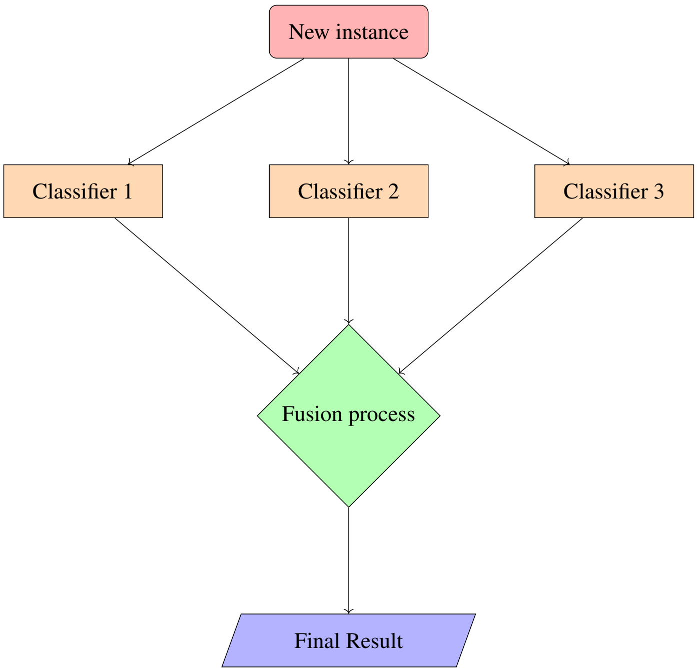


Figure 1.1: Ensemble classifier system

The process of designing optimal ensembles is still under study. However, several factors enable to construct robust ensemble classifiers. The main factors are:

- Classifier generation: Diversity between classifiers is a substantial element for making efficient ensemble systems. It may be achieved by diversifying the input data, the outputs or even the models.

- Classifier combination: The strategy of combining the prediction yielded by each individual classifier.
- Ensemble size: Another main issue of ensemble systems is the number of classifiers which should be merged to get the final decision.

A detailed study of each of these key elements is given in what follows.

1.2 Classifier generation

The first step of classifier ensemble consists of generating a pool of mutually complementary individual classifiers that are accurate and diverse as much as possible. The concept of diversity is relied on the fact that diverse classifiers can make different errors on new instances to be classified. It has been proven that the correlation reduction between individual classifiers increases the accuracy of the ensemble. Diversity can be ensured by using either implicit or explicit techniques (Brown, Wyatt, Harris, & Yao, 2005). The former ones rely on randomness to generate individual diverse classifiers, while the latter ones consists of optimizing a diversity metric. Measuring diversity is still a challenging research topic which has led to the introduction of several diversity measures (Kuncheva & Whitaker, 2003). In what follows, we point out some existing diversity metrics and we discuss the existing procedures allowing to ensure diversity in a classifier ensemble.

1.2.1 Diversity measures

As previously mentioned, diversity is regarded as a vital necessity for the ultimate success of an ensemble classifier. Several diversity metrics have been proposed to compute the diversity between each pair in the ensemble (Kuncheva & Whitaker, 2003). Suppose that $C = \{C_1, \ldots, C_M\}$ be a set of M classifiers. Let $X = \{x_1, \ldots, x_N\}$ be a dataset made up of N objects x_i that are characterized by class labels $y_i \in \Theta = \{\theta_1, \ldots, \theta_c\}$ (i.e $i \in \{1, N\}$ and c is the number of all possible class labels). Each classifier $C_m \in C$ takes as input a query instance z with a class label θ_j (i.e. $m \in \{1, M\}$ and $j \in \{1, c\}$). Let s^m be a vector reflecting the decision of the classifier C_m:

$$s^m = \begin{cases} 1 \text{ if } C_m(z){=}\theta_j \\ 0 \text{ otherwise} \end{cases}$$

The difference between a pair C_m and C_k can then be represented by a correlation analysis matrix (see Table 1.1) that displays the frequency distribution of decisions between C_m and C_k over the given data set X where N_{TT} involves the number of query instances $z \in X$ that are correctly classified through both C_m and C_k, N_{FF} implies the number of query instances $z \in X$ that are incorrectly classified by both C_m and C_k, N_{TF} refers to the number of query instances $z \in X$ that are correctly classified by the classifier C_k and incorrectly classified by the classifier C_m, and N_{FT} represents the number of query instances $z \in X$ that are well classified by the classifier C_m and misclassified by the classifier C_k.

Table 1.1: Distribution of decisions between two classifiers

C_m C_k	$s^m = 1$	$s^m = 0$
$s^k = 1$	N_{TT}	N_{TF}
$s^k = 0$	N_{FT}	N_{FF}

Table 1.1 allows to identify some diversity measures between two given individual classifiers. By the following, we describe the Q-statistic, the correlation coefficient, the disagreement and the double-fault measures.

The Q-statistic

Yule's Q-statistic is one among various statistics for assessing the similarity between the predictions of two classifiers. Given two classifiers C_k and C_m, the Q-statistic measure is given as follows (Crump, 1982):

$$Q_{m,k} = \frac{N_{TT}N_{FF} - N_{FT}N_{TF}}{N_{TT}N_{FF} + N_{FT}N_{TF}} \tag{1.1}$$

It is worth noting that the Q values vary between -1 and 1. The value of 0 stands for the case of statistically independent classifiers. A positive value of $Q_{m,k}$ reflects the case that the classifiers tend to recognize correctly a great number of instances. A negative value of $Q_{m,k}$ refers to the case where both classifiers commit errors on different instances.

Given a set of M classifiers, the averaged Q-statistic over all pairs of classifiers is set to:

$$Q_{avg} = \frac{2}{M(M-1)} \sum_{m=1}^{M-1} \sum_{k=m+1}^{M} Q_{m,k} \tag{1.2}$$

The correlation coefficient ρ

The correlation coefficient between two classifiers C_m and C_k is defined as follows:

$$\rho_{m,k} = \frac{N_{TT}N_{FF} - N_{FT}N_{TF}}{\sqrt{(N_{TT}+N_{TF})(N_{FT}+N_{FF})(N_{TT}+N_{FT})(N_{TF}+N_{FF})}} \tag{1.3}$$

Note that the correlation coefficient measure may also has a positive or a negative value. It is probably best to use the Q-statistic correlation measure against the ρ measure thanks to its simplicity (Kuncheva, 2000).

The disagreement measure

The disagreement measure, introduced by Skalak (1996), has been used to measure the diversity between a base classifier and a complementary one. It has been used then by Ho (1998) to compute the diversity in decision forests. It is set to:

$$Dis_{m,k} = \frac{N_{FT}+N_{TF}}{N_{TT}+N_{FF}+N_{TF}+N_{FT}} = \frac{N_{FT}+N_{TF}}{N} \tag{1.4}$$

The double-fault measure

The double-fault measure, proposed by Giacinto and Roli (2001), has been used to construct an $M \times M$ diversity matrix that contains the degree of dissimilarity between all pairs of classifiers. The aim is to pick out the low correlated classifiers. It is computed as the proportion of the instances that have been misclassified by both classifiers as follows:

$$DF_{m,k} = \frac{N_{FF}}{N_{TT}+N_{FF}+N_{TF}+N_{FT}} = \frac{N_{FF}}{N} \tag{1.5}$$

Note that for all pairwise measures, we have to compute the average diversity yielded by all pairs of classifiers in a given pool such as in Equation 1.2. Note that these pairwise measures have been proposed as measures of similarity, dissimilarity or correlation. Table 1.2 summarizes the presented diversity measures where the arrow indicates whether diversity is greater if the measure is lower (↓) or greater (↑) and *s*, *dis* and *c* reflect respectively similarity, dissimilarity and correlation.

Table 1.2: Pairwise diversity measures

Measure	**↑/↓**	**s/dis/c**	**Range**
Q-statistic	↓	s/c	[-1,1]
Correlation coefficient	↓	s/c	[-1,1]
Disagreement	↑	dis	[0,1]
Double-fault measure	↓	s	[0,1]

1.2.2 Ensuring diversity

As already mentioned, the diversity of a classifier ensemble can be ensured by varying either the input data (training samples, features), the outputs or the models (Giacinto, Roli, & Fumera, 2000; Kuncheva, 2004). By the following, we present in more details each of these approaches.

Manipulating the input data

Both theoretical and practical studies have shown that diversity can be ensured by trained classifiers on different input subspaces. Two main techniques have been considered: the manipulation of the training samples and the manipulation of the training features.

Diversifying training data It consists of training a learning algorithm on different subsets of the original training data. This technique has shown a great success especially when dealing with unstable classifiers (e.g. neural networks and decision trees). Examples include Bagging (Breiman, 1996) and Boosting (Freund, 1995). Bagging performs random sampling with replacement in order to get independent training subsets. The boosting approach updates the training samples weights while considering the misclassified samples yielded by the previous classifiers. The classifiers obtained are then aggregated, using the majority vote operator.

Diversifying training features Data features can also be used to ensure the diversity of a pool of classifiers. One of the well-known algorithm is the random subspacing (Ho, 1995, 1998). It has been used by several machine learning classifiers, including decision trees (Breiman, 2001), SVM (Lienemann, Plötz, & Fink, 2007) and linear classifiers (Skurichina & Duin, 2002). Attribute bagging is another method for ensuring diversity. It creates random projections of a given training set by a random selection of feature subsets (Bryll et al., 2003). Feature subset selection based on relevance-based algorithm has also been introduced and has yielded satisfactory results (Bell & Wang, 2000). The rough set theory, proposed in (Pawlak, 1998), has also been used to construct diverse classifiers. This approach generate firstly all possible minimal subset of attributes (i.e. called reducts) allowing the same classification ability as the original attribute set. Then, a diverse subset of these reducts has to be used for training an ensemble of individual classifiers (Hu, Yu, Xie, & Li, 2007; Debie, Shafi, Lokan, & Merrick, 2013).

Manipulating outputs

In this approach, diversity can be ensured by diversifying the individual classifier outputs. That is, each classifier has the authority to classify some classes, especially for handling multi-class classification problems. Error Correcting Output Coding (ECOC) that has been developed by Dietterich and Bakiri (1995) is a well known example. It uses a code matrix to transform a multi-class classication problem into multiple binary ones. The binary classifiers have be merged for yielding the final decision (Dietterich & Bakiri, 1995). Switching label is another technique for handling outputs (Martínez-Muñoz & Suárez, 2005). It aims to produce an ensemble of classifiers trained on perturbed versions of the training set where the training classes are randomly switched.

Diversifying models

A further method for achieving diversity consists of diversifying the classification models. Different versions of the same learning algorithm can be more effective than using different machine leaning algorithms. One alternative solution consists of injecting randomness into the learning algorithm. For instance, different initial weights can be randomly assigned to neural networks. The resulting classifiers, using the same training data but different initial weights, can be quite diverse. Dietterich (2000) has suggested an ensemble of decision trees by introducing randomness when selecting the best splitting attributes at each internal node.

1.3 Classifier combination

The process of combining the output predictions of a pool of individual classifiers constitutes a fundamental step for any ensemble classifier. The combination strategy depends mainly on the classifier output forms. Two output forms can be distinguished: single class label and continuous outputs. In this Section, we outline some combination rules belonging to each output level.

1.3.1 Combining class labels

This kind of combination rules is dedicated to classifiers providing specific class support. Assume that $\Theta = \{\theta_1, \theta_2, \ldots, \theta_c\}$ be a set of c classes and let $C = \{C_1, C_2, \ldots, C_M\}$ be a set of M classifiers. Suppose that z is a query pattern that has to be classified into one class of Θ. The classifier decisions $d_{i,j}$ yielded by a classifier C_i for a class label θ_j (i.e. $i \in \{1, \ldots, M\}$ and $j \in \{1, \ldots, c\}$) can be represented in a binary format as follows:

$$d_{i,j}(z) = \begin{cases} 1 & \text{if } C_i(z)=\theta_j \\ 0 & \text{otherwise} \end{cases} \tag{1.6}$$

The decision $d_{i,j}(z)$ equals 1 if θ_j corresponds to the the predicted class of z through the classifier C_i. Otherwise, it is equal to 0.

Majority voting

The majority voting allows to assign a query pattern to the class with the largest number of votes. Depending to the ensemble decision, three main versions can be distinguished:

- **Unanimous voting:** This case requires that all classifiers are agreed.
- **Simple majority:** The prediction is given by at least one more than half the pool of classifiers.
- **Majority voting:** the prediction is assigned to the class with the highest number of votes.

The decision class $\theta_{j^*}(z)$ of z using the majority voting operator can be mathematically defined as follows:

$$\theta_{j^*}(z) = argmax_{j\in\{1,\ldots,c\}} \sum_{i=1}^{M} d_{i,j}(z) \tag{1.7}$$

Weighted Majority Voting (WMV)

This aggregation rule, proposed by Littlestone and Warmuth (1994), is used in the case where some classifiers are likely to be correct than others. The main idea behind the WMV rule is to assign a weight w_i for each individual classifier C_i in proportion of its estimated performance. The decision class corresponds to θ_{j^*} only if θ_j receives the greatest total weighted vote:

$$\theta_{j^*}(z) = argmax_{j\in\{1,\ldots,c\}} \sum_{i=1}^{M} w_i \times d_{i,j}(z) \tag{1.8}$$

The process of assigning weights still an open question. One commonly strategy relies on the performance of each individual classifier on either a validation set or a training set as an estimation of the generalization performance.

Behavior Knowledge Space (BKS)

This rule uses a look up table to estimate the posterior probabilities and every combination of votes (Huang & Suen, 1993). Let M be a set of individual classifiers for solving a c-class classification problem. The BKS table contains c^M

entries (i.e. all possible combinations) where each one keeps the distribution of c true labels in the training set. Table 1.3 illustrates an example of the BKS table with three class labels $\Theta = \{\theta_1, \theta_2, \theta_3\}$ and two classifiers $C = \{C_1, C_2\}$. Cells below C_1 and C_2 correspond to all possible predictions. Entries below true class represent the distribution of the true labels that the training data fall into. Suppose that the predictions of the classifier C_1 and C_2, for a query instance z, are respectively θ_1 and θ_1. The final class label of z is obtained by identifying the index that corresponds to the combination $C_1(z) = \theta_1$ and $C_2(z) = \theta_1$ from the look up table. From this, the class label of z is θ_1.

Table 1.3: An example of the BKS table

Prediction		True class			Chosen Class
C_1	C_2	θ_1	θ_2	θ_3	
θ_1	θ_1	**10**	3	3	θ_1
θ_1	θ_2	3	0	**6**	θ_3
θ_1	θ_3	**5**	4	0	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
θ_3	θ_2	2	2	**5**	θ_3
θ_3	θ_3	0	1	**6**	θ_3

Borda count

Borda count, originally introduced in 1770 by Jean-Charles de Borda, is regarded as one among the most common classifier combination rules. It considers each classifier as a voter and the classes as the candidates. It computes at first a preference ranking from all voters over all candidates and then it sums the rankings relative to each class. The query instance has to be assigned to the class with the highest sum of votes. Table 1.4 presents a Borda count example using three classifiers denoted respectively by C_1, C_2 and C_3 and four class labels $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4\}$.

The borda count results is given as follows:

- $R_{\theta_1} = 4 + 3 + 1 = 8$

Table 1.4: Borda count example

Rank value	C_1	C_2	C_3
4	θ_3	θ_1	θ_2
3	θ_2	θ_2	θ_1
2	θ_4	θ_4	θ_3
1	θ_1	θ_3	θ_4

- **R_{θ_2} = 3+3+4 = 10**
- R_{θ_3}= 4+2+1= 7
- R_{θ_4} = 2+2+1 = 5

According to these results, the query pattern will be assigned to the class θ_2 as it has the highest ranking sum.

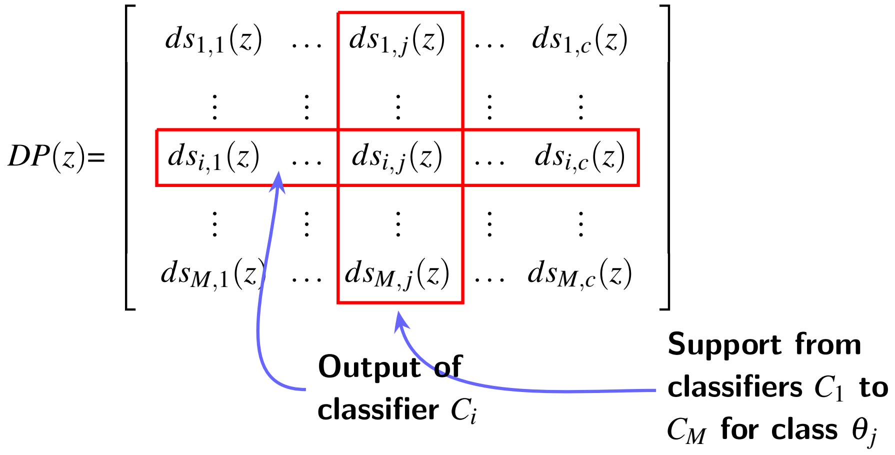
1.3.2 Combining continuous outputs

In what follows, we pointed out some fusion techniques for classifiers providing continuous outputs. That is, every individual classifier assigns a degree of support for each class θ_j in Θ that can be interpreted as an estimation of its posterior probability under the circumstance that the supports over all classes should equal 1. Kuncheva, Bezdek, and Duin (2001) have introduced the decision profile matrix $DP(z)$ that stores the outputs of M classifiers relative of a query instance z. The decision profile matrix, is depicted in Figure 1.2 where $ds_{i,j}(z)$ corresponds to the degree of support relative to the class θ_j through the classifier C_i.

The whole support committed exactly to the class θ_j is obtained by combining the individual classifiers. Several combination rules have been introduced and described by the following:

The average operator

It is regarded as one of the simplest algebraic combiners. The degree of support assigned to the class θ_j is computed as the average of all classifiers' supports for



$$DP(z)=\begin{bmatrix} ds_{1,1}(z) & \dots & ds_{1,j}(z) & \dots & ds_{1,c}(z) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ ds_{i,1}(z) & \dots & ds_{i,j}(z) & \dots & ds_{i,c}(z) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ ds_{M,1}(z) & \dots & ds_{M,j}(z) & \dots & ds_{M,c}(z) \end{bmatrix}$$

Output of classifier C_i
Support from classifiers C_1 to C_M for class θ_j

Figure 1.2: Decision profile matrix

that class. It is to:

$$Sup_{\theta_j}(z) = \frac{1}{M}\sum_{i=1}^{M} ds_{i,j}(z) \tag{1.9}$$

The class with the highest support will be selected as the predicted class of the query instance. An example of this combination operator is given in Figure 1.3. In this example, the query pattern is assigned to the class θ_3 as it has the greatest support.

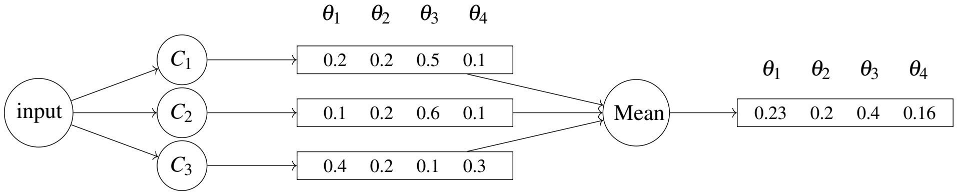

θ_1 θ_2 θ_3 θ_4
input
C_1
C_2
C_3
0.2 0.2 0.5 0.1
0.1 0.2 0.6 0.1
0.4 0.2 0.1 0.3
Mean
θ_1 θ_2 θ_3 θ_4
0.23 0.2 0.4 0.16

Figure 1.3: Fusion operator example

The K trimmed mean

In some cases, classifiers may give extreme degrees of support. Trimmed mean is designed to cope with this problem. In fact, it allows to exclude the $K\%$ largest and the $K\%$ smallest support degrees and it considers the arithmetic mean of the remaining support degrees.

The minimum/maximum rule

These rules depend mainly on the classifier that has the minimum/maximum degree of support for each class. Concerning the maximum rule, the chosen class is the one with the highest support degree:

$$\theta_{j^*}(z) = argmax_{j \in \{1,\dots,c\}} SuppMax_{\theta_j}(z) \tag{1.10}$$

where

$$SuppMax_{\theta_j}(z) = \max_{i=1}^{M}\{ds_{i,j}(z)\} \tag{1.11}$$

For the minimum rule, it consists of selecting the class receiving the lowest degree of support:

$$\theta_{j^*}(z) = argmin_{j \in \{1,\dots,c\}} SuppMin_{\theta_j}(z) \tag{1.12}$$

$$SuppMin_{\theta_j}(z) = \min_{i=1}^{M}\{ds_{i,j}(z)\} \tag{1.13}$$

The sum rules

This rule sums the degree of support yielded by each individual classifier and assigns the test pattern to the class with the highest support.

The product rule

This combination operator requires the assumption that the combined classifiers are mutually independent. In this method, the degrees of support yielded by the individual classifiers, for each class θ_j, are multiplied:

$$Supp_{\theta_j}(z) = \frac{1}{M}\prod_{i=1}^{M} ds_{i,j}(z) \tag{1.14}$$

This method is very sensitive to the degree of support that is close to zero or even to smallest supports.

1.4 Ensemble size

Another substantial key element when designing an ensemble classifier is the ensemble size. There is no doubt that a huge number of classifiers may in the one hand increase the computational complexity and on the other hand decrease the comprehensibility. Several researches have been done to predefine a reasonable ensemble classifier size. Hansen and Salamon (1990) claim that ensembles of ten classifiers are sufficient for reducing the error rate. Further experimentations, for neural networks and decision trees ensembles using up to 100 classifiers have been performed by Opitz and Maclin (1990). Numerical results have shown that the reduction in error, for both Bagging and Boosting applied to neural networks, have occurred after 10 to 15 classifiers. For the case of AdaBoost decision tree, the error reduction is obtained for ensembles containing 25 classifiers. The conclusion conducted following this study proves that ensembles of 25 classifiers are sufficient for reducing the error rate and consequently to improve the classification performance.

1.5 Conclusion

This Chapter is devoted to highlighting the fundamental concepts of an ensemble classifier. Three main issues when designing a classifier ensemble have been discussed. The first one, which is classifier generation, is focused on strategies for constructing diverse individual classifiers either by manipulating the input data, the outputs or the models. The second issue is about the strategies used for combining ensemble classifiers. Two main strategies have been distinguished: class label outputs and continuous output. Concerning the third issue, it concerns the number of individual classifiers that have to be merged to achieve good performance.

One should note that individual classifier outputs can be modeled as a fuzzy membership function, evidential basic belief assignment function to cite a few. To do so, fuzzy rules (Cho & Kim, 1995a, 1995b) and belief function techniques (Franke & Mandler, 1992; Rogova, 1994; Tresp & Taniguchi, 1995) are used. Ensemble classifiers within the belief function framework will be discussed in more details in the next Chapter.

Chapter 2

Ensemble classifiers within the belief function framework

Contents

2.1 Introduction 22

2.2 Basic concepts of the belief function theory 22

  2.2.1 Knowledge representation 23

  2.2.2 Special bbas 25

  2.2.3 Discounting 27

  2.2.4 The dissimilarity between two pieces of evidence . . . 27

  2.2.5 Combining information sources 28

  2.2.6 Decision process 32

2.3 Ensemble systems within the belief function theory 33

  2.3.1 Mandler and Shurmann method 34

  2.3.2 Xu et al.'s method 35

  2.3.3 Rogova's method 36

  2.3.4 Al-Ani's method 37

  2.3.5 The calibration method 37

2.4 Conclusion 38

2.1 Introduction

Ensemble classifiers have rapidly become popular techniques for improving the performance of complex classification problems. One of the main issues in ensemble systems is that individual classifiers may predict the label class of objects with some uncertainty. Up to now, various approaches have been proposed to dealing with knowledge imperfection, including the fuzzy theory (Umano et al., 1994), the probability theory (Quinlan, 1987), the possibilistic theory (Hüllermeier, 2002; Jenhani, Elouedi, BenAmor, & Mellouli, 2005; Jenhani, BenAmor, & Elouedi, 2008) and the belief function theory (Elouedi, Mellouli, & Smets, 2001; Vannoorenberghe, 2004; Vannoorenberghe & Denœux, 2002). This latter approach has been proven to be a valuable tool for representing and managing the uncertainty associated with the classifier outputs (Mandler & Shurmann, 1988; L. Xu, Krzyżak, & Suen, 1992; Rogova, 1994; Al-Ani & Deriche, 2002; Denœux, 1995). The main advantage that makes the belief function theory very appealing over the other existing theories, is its ability to express in a flexible way all kinds of information availability: certain case, partial ignorance and the total ignorance.

In this Chapter, we provide at first an overview of the fundamental concepts of the belief function theory, including knowledge representation, decision making, knowledge combination, etc. Besides, we emphasize some of the best-known researches involving ensemble classifiers within the belief function framework.

2.2 Basic concepts of the belief function theory

The belief function theory, also referred to as evidence theory, is regarded as a very effective and efficient way for representing and managing uncertain knowledge. It was at first introduced by Dempster (1967) within the context of statistical inference. It was then formalized by Shafer (1976) into a generic framework for modeling epistemic uncertainty and developed by Smets (1990) under the name of Transferable Belief Model (TBM).

It is important to underline that this theory is extensively used for handling several real-world applications, including image processing (Bloch, 1996; Lefèvre, Colot, & Vannoorenberghe, 2002), business decision (Srivastava & Mock, 2002), multi sensor fusion (H. Kim & Swain, 1995; Appriou, 1999), pattern recognition (Tupin, Bloch, & Maître, 1999; Denœux & Zouhal, 2001), medical diagnosis

(Smets, 1981; Straszecka, 2006), classification (Elouedi et al., 2001; Trabelsi, Elouedi, & Lingras, 2011) and target tracking (Daum, 1996), etc.

In this Section, we firstly provide a brief overview of the fundamental concepts of the belief function theory as interpreted by the TBM framework. Besides, we point out other basic concepts such as the special belief functions, the discounting process, the decision making and the dissimilarity degree between two bbas. Then, we outline some of the well commonly used combination rules for merging both distinct and non-distinct information sources.

2.2.1 Knowledge representation

Frame of discernment

Let $\Theta = \{\theta_1, \theta_2, \ldots, \theta_c\}$ denotes the frame of discernment including a finite non empty set of c elementary hypothesis that are assumed to be exhaustive and mutually exclusive. The power set of Θ, denoted by 2^Θ, is made up of all the subsets of Θ:

$$2^\Theta = \{\emptyset, \{\theta_1\}, \{\theta_2\}, \{\theta_1, \theta_2\}, \ldots, \Theta\} \tag{2.1}$$

Each element of 2^Θ is called a proposition or an event.

Basic belief assignment

An expert's belief over the subsets of the frame of discernment Θ are represented by the so-called basic belief assignment (bba) denoted by m. It is carried out in the following manner:

$$\sum_{A \subseteq \Theta} m(A) = 1 \tag{2.2}$$

The basic belief mass (bbm), denoted by $m(A)$, implies the degree of belief exactly assigned to the event A. Because of a lack of information, this quantity cannot be distributed to any strict subset of A. It is worth noting that every subset A of 2^Θ having fulfilled $m(A) > 0$ is called a focal element.

In his original work, Shafer (1976) has assigned a null value to the empty set (i.e., impossible proposition). Such a bba is commonly known as a normalized basic belief assignment. However, Smets (1990) has introduced the concept of unnormalized belief functions ($m(\emptyset) \neq 0$) within the TBM framework where $m(\emptyset)$ has been interpreted either as the conflict amount between pieces of evidence or as the part of evidence given when the true value does not belong to Θ.

As well, it is quite possible to transform any unnormalized belief function into a normalized one as follows:

$$m(A) = \begin{cases} 0 & \text{if } A = \emptyset \\ k.m(A) & \text{otherwise} \end{cases} \tag{2.3}$$

where k^{-1} = 1-$m(\emptyset)$ reflects the normalization factor.

Belief function

A belief function *bel*, relative to a bba *m*, assigns to any subset *A* of Θ the sum of beliefs exactly committed to every subset of *A* by *m* (Shafer, 1976). In other words, it implies the total belief that one commits to *A* without being also committed to $\overline{A}$. It is worth noting that $m(\emptyset)$ is not included in $bel(A)$, since $\emptyset$ is a subset of both *A* and $\overline{A}$. The belief function *bel* is defined as follows:

$$\begin{aligned} bel &: 2^{\Theta} \rightarrow [0,1] \\ bel(A) &= \sum_{\emptyset \neq B \subseteq A} m(B) \end{aligned} \tag{2.4}$$

Plausibility function

The plausibility function $pl(A)$ stipulates the largest possible support that could be assigned to a subset *A* of Θ. It is calculated as the sum of the bbms relative to subsets *B* compatible with *A* (i.e., do not contradict *A*). The plausibility function is defined as follows (Barnett, 1991):

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

$$pl(A) = \sum_{A \cap B \neq \emptyset} m(B) \tag{2.5}$$

$$= bel(\Theta) - bel(\overline{A}) \tag{2.6}$$

Where $\overline{A}$ corresponds to the complement of the subset *A* relative to Θ.

Commonality function

The commonality function $q(A)$ quantifies the total mass that can move freely to any element of a proposition A (Barnett, 1991). It is equal to the sum of the masses allocated to the supersets of A (i.e., having A in common). It is set as follows:

$$
\begin{aligned}
q &: 2^{\Theta} \rightarrow [0,1] \\
q(A) &= \sum_{B \supseteq A} m(B) && (2.7) \\
& && (2.8)
\end{aligned}
$$

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

2.2.2 Special bbas

Herein, we point out the special cases of bbas, namely vacuous bba, categorical bba, Bayesian bba, simple support function, consonant bba, certain bba, dogmatic and non−dogmatic bbas.

Vacuous bba

A vacuous bba refers to a bba having Θ as its unique focal element. Such a case reflects the state of total ignorance. A vacuous bba is then defined as follows:

$$m(\Theta) = 1 \text{ and } m(A) = 0 \; \forall \, A \neq \Theta \tag{2.9}$$

Categorical bba

A bba is called categorical where it has a unique focal element A. It is defined as follows:

$$m(A) = 1, \text{ for some } A \subset \Theta \tag{2.10}$$

$$m(B) = 0, \text{ for } B \subseteq \Theta, B \neq A \tag{2.11}$$

Bayesian bba

A bba is called Bayesian where all of its focal elements are singletons:

$$\text{if } m(A) > 0 \text{ then } |A| = 1 \tag{2.12}$$

Simple support function

A bba is called simple support function (ssf) if it has at most two focal elements: the frame of discernment Θ and a strict subset of Θ called the focus of the ssf (Smets, 1995). A simple support function is defined as:

$$m(X) = \begin{cases} w & \text{if } X = \Theta \\ 1-w & \text{if } X = A \text{ for some } A \subseteq \Theta \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

where A is the focus and $w \in [0,1]$.

In simple terms, such ssf can be written as A^w where A is the focus and $w \in [0,1]$.

Consonant bba

A bba is called consonant if its focal elements (A_1,A_2,...,A_n) are nested:

$$A_1 \subseteq A_2 \subseteq ... \subseteq A_n \tag{2.14}$$

Certain bba

A certain bba is a particular case of the categorical belief function where its unique focal element is a singleton. This bba represents the state of total certainty and it is defined as follows:

$$m(A) = 1 \text{ for some } A \subset \Theta \text{ and } |A|=1 \tag{2.15}$$

and

$$m(B) = 0 \text{ for all } B \subseteq \Theta \text{ and } B \neq \text{A} \tag{2.16}$$

Dogmatic and non dogmatic bba

Dogmatic bba A bba is called dogmatic if the frame of discernment Θ is not a focal element:

$$m(\Theta) = 0 \tag{2.17}$$

Non-dogmatic bba A bba is called non-dogmatic if the frame of discernment Θ is a focal element (Smets, 1995):

$$m(\Theta) > 0 \tag{2.18}$$

2.2.3 Discounting

The reliability of each expert can be quantified. In fact, if experts are not fully reliable a method of discounting seems imperative to update experts' beliefs. Let $m(A)$ be a bba induced from an information source S with a reliability rate $1-\alpha$. The discounted bba $m^{\alpha}(A)$ is obtained as follows:

$$\begin{aligned} m^{\alpha}(A) &= (1-\alpha)m(A) \text{ for } A \subset \Theta. \\ m^{\alpha}(\Theta) &= \alpha + (1-\alpha)m(\Theta) \end{aligned} \tag{2.19}$$

where α reflects the discounting factor.

2.2.4 The dissimilarity between two pieces of evidence

In the research literature, several measures have been proposed to compute the dissimilarity between two given bbas (A.-L. Jousselme & Maupin, 2012; A. Jousselme, Grenier, & Bossé, 2001; Ristic & Smets, 2006; Tessem, 1993). One of the earliest and the best-known measures is the Jousselme distance. Formally, the Jousselme distance, for two given bbas m_1 and m_2, is defined as (A. Jousselme et al., 2001):

$$d(m_1, m_2) = \sqrt{\frac{1}{2}(m_1 - m_2)^T D(m_1 - m_2)} \tag{2.20}$$

where the Jaccard similarity measure D is set to:

$$D(A,B) = \begin{cases} 1 & \text{if } A{=}B{=}\ \emptyset \\ \dfrac{|A \cap B|}{|A \cup B|} & \forall\, A,B \in 2^{\Theta} \end{cases} \tag{2.21}$$

2.2.5 Combining information sources

The fusion of imperfect data is a crucial task owing to its ability to achieve a more accurate information and improve decision making. The belief function theory is regarded as a powerful tool to merge imperfect knowledge (uncertain, imprecise and incomplete data) (Lefèvre et al., 2002; Klein, Lecomte, & Miche, 2008).

Indeed, several combination rules have been proposed to aggregate a set of evidence induced from different information sources. Some of these rules deal with independent information sources (Dempster, 1967; Dubois & Prade, 1986; Yager, 1987; Smets, 1998; Martin, 2012), whereas others assume information sources combined to be distinct (Cattaneo, 2003; Denœux, 2006, 2008; Boubaker, Elouedi, & Lefèvre, 2013). In what follows, we present some combination rules dealing with both independent and non-independent information sources.

Combining independent information sources

Let m_1 and m_2 be two bbas induced from two independent information sources and defined in the same frame of discernment Θ. Several combination rules have been proposed to combine such kind of bbas. Ones among the most commonly combination operators are the conjunctive and the disjunctive rules (Smets, 1998), the Dempster rule (Dempster, 1967) and the Combination With Adapted Conflict (CWAC) rule (Lefèvre & Elouedi, 2013):

Conjunctive rule The conjunctive rule of combination, proposed by Smets (1998), is used to combine two bbas provided by reliable and distinct information sources. The resulting bba, denoted by $m_1 \bigcirc\!\!\!\!\cap\, m_2$, is defined by:

$$(m_1 \bigcirc\!\!\!\!\cap\, m_2)(A) = \sum_{B,C \subseteq \Theta : B \cap C = A} m(B).m(C) \tag{2.22}$$

The conflict, denoted by $m_1 \bigcirc\!\!\!\!\!\cap\, m_2(\emptyset)$, quantifies the degree of disagreement between the two combined sources.

Note that the conjunctive rule can be expressed in terms of the commonality functions as follows:

$$(q_1 \bigcirc\!\!\!\!\!\cap\, q_2)(A) = q_1(A)q_2(A) \tag{2.23}$$

where q_1 and q_2 denote respectively the commonality functions of m_1 and m_2.

Dempster's rule The Dempster rule, called also the orthogonal sum, constitutes the normalized version of the conjunctive rule where the mass assigned to the empty set must be reallocated over all focal elements thanks to a normalization factor k (Shafer, 1976). As the conjunctive rule, the Dempster operator assumes the combined pieces of evidence to be reliable and distinct. It is set to:

$$(m_1 \oplus m_2)(A) = k(m_1 \bigcirc\!\!\!\!\!\cap\, m_2)(A) \tag{2.24}$$

and

$$(m_1 \oplus m_2)(\emptyset) = 0 \tag{2.25}$$

where

$$k^{-1} = 1 - (m_1 \bigcirc\!\!\!\!\!\cap\, m_2)(\emptyset) \tag{2.26}$$

Disjunctive rule The disjunctive combination rule, which is the dual of the conjunctive rule, is used to combine two bbas m_1 and m_2 when at least one of them is fully reliable, but we do not know which one. This rule is defined as follows (Smets, 1998):

$$(m_1 \bigcirc\!\!\!\!\!\cup\, m_2)(A) = \sum_{B,C \subseteq \Theta : B \cup C = A} m_1(B).m_2(C) \tag{2.27}$$

Combination With Adapted Conflict rule (CWAC) As we have already explained, the mass function $m(\emptyset)$, resulting of the conjunctive combination rule, reflects the degree of conflict between the combined sources. However, the basic belief mass assigned to the empty set tends toward 1 when we apply a large number of conjunctive combinations (Lefèvre & Elouedi, 2013). Therefore, the conflict loses its original role as an alarm signal indicating that there is a sort of disagreement between sources. To cope with this shortcoming, Lefèvre and Elouedi (2013) have suggested the CWAC rule which is able to maintain the conflict as an alarm signal when combining sources.

The CWAC combination rule is defined by an adaptive weighting between the conjunctive and the Dempster rules. This adaptive weighting provides an effective way to obtain the same behavior as the conjunctive rule when belief functions are contradictory and the same behavior as the Dempster rule when belief functions are similar (Lefèvre & Elouedi, 2013).

So, a distance measure must be used for calculating the dissimilarity d between two information sources. Note that the CWAC rule relies preliminary on the Jousselme distance (see Equation 2.20):

- If $d(m_1, m_2) = 0$ then m_1 and m_2 are in agreement and their combination should not generate a conflict. Consequently, the conflict must be redistributed in the same manner as the Dempster rule.
- If $d(m_1, m_2) = 1$ then m_1 and m_2 are in disagreement and their combination generate a conflictual mass which must be kept in the same way as the conjunctive combination rule.

The CWAC rule, denoted by $\textcircled{\leftrightarrow}$, is defined as:

$$m_{\textcircled{\leftrightarrow}}(A) = \gamma_1 m_{\textcircled{\cap}}(A) + \gamma_2 m_{\oplus}(A) \tag{2.28}$$

with:

$$m_{\oplus}(A) = (m_1 \oplus m_2)(A) \ \forall\, A \subseteq \Theta \tag{2.29}$$

$$m_{\textcircled{\cap}}(A) = (m_1 \textcircled{\cap} m_2)(A) \ \forall\, A \subseteq \Theta \tag{2.30}$$

where γ_1 and γ_2 should satisfy the following conditions:

$$\gamma_1 + \gamma_2 = 1 \tag{2.31}$$

with :

$$\gamma_1 = d(m_1, m_2); \gamma_2 = 1 - d(m_1, m_2) \tag{2.32}$$

Dependent combination rules

As mentioned in the beginning of this Section, the belief function theory provides several combination rules where some of them handle independent information sources while others deal only with dependent information sources. Herein, we present some well-known dependent combination rules, notably the cautious conjunctive rule and its normalized version (Denœux, 2006) and the cautious CWAC rule (Boubaker et al., 2013).

Cautious conjunctive rule The cautious conjunctive rule, denoted by $\owedge$, has been proposed by Denœux (2006) to aggregate pieces of evidence induced from reliable dependent information sources using the conjunctive canonical decomposition stated by Smets (1995). Let m_1 and m_2 be two non-dogmatic bbas, the result of their combination, denoted by $m_1 \owedge m_2$, is given as follows (Denœux, 2006):

$$m_1 \owedge m_2(A) = \textcircled{\cap}_{A \subset \Theta} A^{w_1(A) \wedge w_2(A)} \tag{2.33}$$

where $w_1(A) \wedge w_2(A)$ corresponds to the weight function of a bba $m_1 \owedge m_2$ and $\wedge$ represents the minimum operator. The weights $w(A)$ for every $A \subset \Theta$ can be obtained from the commonalities as follows:

$$w(A) = \prod_{B \supseteq A} q(B)^{(-1)^{|B|-|A|-1}}. \tag{2.34}$$

One of the crucial drawbacks of the cautious conjunctive rule is its inability to preserve the main role of the conflict.

Normalized cautious rule The normalized version of the cautious conjunctive rule is obtained by replacing the conjunctive operator $\textcircled{\cap}$ by the Dempster operator $\oplus$ to overcome the conflict effect (Denœux, 2006). This rule is then defined as:

$$m_1 \owedge^* m_2 = \bigoplus_{\emptyset \neq A \subset \Theta} A^{w_1(A) \wedge w_2(A)} \tag{2.35}$$

We thus have:

$$m_1 \owedge^* m_2(A) = k.m_1 \owedge m_2(A) \quad \text{and} \quad m_1 \owedge^* m_2(\emptyset) = 0 \tag{2.36}$$

with $k^{-1} = 1 - m_1 \owedge m_2(\emptyset)$.

The weight functions of $m_1 \owedge^* m_2$ are calculated as follows:

$$w_1 \owedge^* w_2(A) = w_1 \owedge w_2(A) = w_1(A) \wedge w_2(A), \forall\, A \in 2^{\Theta} \setminus \{\emptyset, \Theta\} \tag{2.37}$$

The cautious combination with Adaptive Conflict The cautious CWAC rule, based on the cautious rule and inspired from the behavior of the CWAC rule, is defined by an adaptive weighting between the unormalized cautious and the normalized one (Boubaker et al., 2013). This rule allows also to preserve the initial role of the conflict as an alarm signal reflecting the disagreement between sources.

The cautious CWAC rule is then defined as follows:

$\forall\, A \subseteq \Theta\,,\, m_{\bigcirc\!\!\!\!\wedge}(\emptyset) \neq 1:$

$$m_{\odot}(A) = d(m_1,m_2)m_{\bigcirc\!\!\!\!\wedge}(A) + (1-d(m_1,m_2))m_{\bigcirc\!\!\!\!\wedge^*}(A) \tag{2.38}$$

2.2.6 Decision process

Decision making aims to select the most reasonable hypothesis for a given problem. In this subsection, we outline some approaches to ensure decision making within the belief function framework. Firstly, we present the most used one which is the pignistic probability; proposed by the Transferable Belief Model (TBM) (Smets, 1988). Then, we detail two other methods, including the maximum of credibility and the maximum of plausibility.

Pignistic probability

According to the TBM framework, holding beliefs and making decisions are two distinct processes. Therefore, it is based on two level models as depicted in Figure 2.1:

- The credal level where beliefs are represented by belief functions.
- The pignistic level where beliefs are transformed into probability functions called the pignistic probabilities denoted by *BetP* in order to make decision. The pignistic probability is computed as follows (Smets, 1988):

$$BetP(A) = \sum_{B \subseteq \Theta} \frac{|A \cap B|}{|B|} \frac{m(B)}{1-m(\emptyset)}\ \forall\, A \in \Theta \tag{2.39}$$

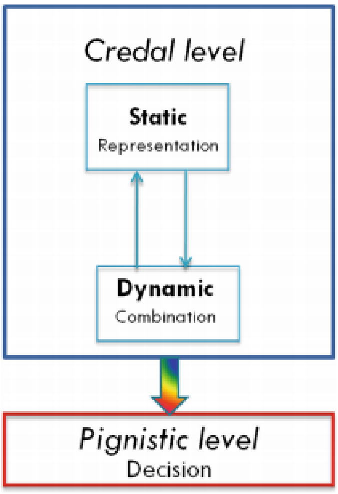


Figure 2.1: The generic structure of the TBM framework

Maximum of credibility

It consists of choosing the singleton event of Θ that has the highest value of the belief function bel. In other words, the most credible hypothesis.

Maximum of plausibility

In analogy with the maximum of credibility, the maximum of plausibility allows to choose the singleton event of Θ that has the greatest value of pl. So that, the most plausible hypothesis should be selected for the final decision (Barnett, 1991).

2.3 Ensemble systems within the belief function theory

Data uncertainty is regarded as one of the main issues of several real world applications that can affect experts' decisions. Within the fields of data mining and pattern recognition, several researches have been carried out to process the uncertainty associated to the classifier outputs, notably by transforming the classifier predictions into evidence. To gain the best performance, ensemble systems within the belief function framework have been well studied for several years now. Figure

2.2 represents the general structure of ensemble systems within the belief function framework.

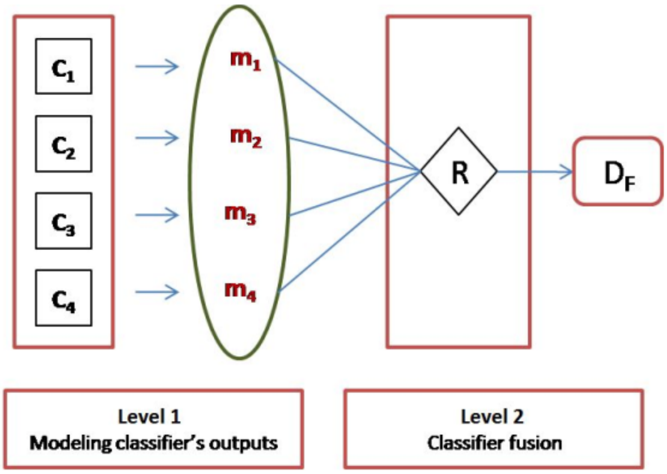


Figure 2.2: General framework of ensemble systems with belief function framework

The following subsections provides a detailed description of some of the best known existing ensemble system techniques within the belief function framework.

2.3.1 Mandler and Shurmann method

One of the preliminary carried out works has been proposed by Mandler and Shurmann (1988). This method consists of transforming the output of each individual classifier into evidence through the use of distance measures. It consists firstly of computing the distance between learning data points and a given number of reference points to assess the statistical distributions of both inter and intra class distances. These distributions, indicating the degree of belonging of an input pattern to a certain reference point, are required to estimate the class-conditional probabilities that has to be converted into evidence and merged. The Dempster rule of combination will then be used to merge the evidence obtained by the different individual classifiers. To sum up, this method requires the choice of the reference points as well as the distance measure. According to Rogova (1994), one must absolutely avoid approximations associated with parameter estimation of statistical models for inter/intra class distances because of its impact on the evidence measure.

2.3.2 Xu et al.'s method

L. Xu et al. (1992) have investigated a belief function model for merging the prediction of multiple classifiers, notably those with single class labels. Their proposed approach transforms the output of a standard classifier into evidence on the basis of its performance. Examples include the recognition rate (correct answer), the substitution rate (wrong answer) and the rejection rate that have to be calculated from a confusion matrix. Let $C = \{C_1, \ldots, C_M\}$ be a set of M classifiers, and let $\Theta = \{\theta_1, \ldots, \theta_c\}$ be a set of c class labels. Each classifier takes as input a pattern test z and outputs either a class label θ_j (i.e. $j \in \{1, \ldots, c\}$) or a rejection class θ_{c+1}. The fundamental form of the confusion matrix for a c-class classification problem is depicted in Figure 2.3 where N_{ij} corresponds to the total number of instance having θ_i as real class labels while there are classified as θ_j.

		Predicted label				
		θ_1	…	θ_j	…	θ_c
Actual label	θ_1	N_{11}	…	N_{1j}	…	N_{1c}
	⋮ θ_i ⋮	N_{i1}	…	⋮ N_{ij} ⋮	…	N_{ic}
	θ_c	N_{c1}		N_{cj}		N_{cc}
	$\theta_{(c+1)}$	$N_{(c+1)1}$		$N_{(c+1)j}$		$N_{(c+1)c}$

Figure 2.3: Confusion Matrix

Let N be the total number of query patterns, the performance rates relative to a given classifier C_k will then be computed as follows:

- The recognition rate:

$$R_k = \frac{\sum_{i=1}^{c} N_{ii}}{N} \tag{2.40}$$

- The substitution rate:

$$S_k = \frac{\sum_{i=1}^{c} \sum_{j=1; i \neq j}^{c} N_{ij}}{N} \tag{2.41}$$

- The rejection rate:

$$T_k = 1 - (R_k + S_k) \tag{2.42}$$

According to L. Xu et al. (1992), the output label θ_j relative to the query pattern z through a classifier C_k (k$\in \{1, \ldots M\}$) will be modeled in terms of evidence as follows:

$$m_j^k(\{\theta_j\}) = R_k, \forall\ \theta_j \in \Theta \tag{2.43}$$

$$m_j^k(\bar{\theta}_j) = T_k, \forall\ \theta_j \in \Theta,\ \bar{\theta}_j = \Theta \setminus \{\theta_i\} \tag{2.44}$$

$$m_j^k(\Theta) = S_k \tag{2.45}$$

The evidence obtained by M classifiers for a given test instance z will be merged by the Dempster rule (Dempster, 1967). The test instance is assigned to the class with the highest support degree.

2.3.3 Rogova's method

Rogova (1994) has discussed a model for merging the outputs of neural network classifiers within the belief function framework. This work is preliminary based on proximity measures (e.g. the cosine function, the Euclidean distance, etc) between a reference vector and the output vector of a given classifier. A reference [2]vector R_i^k has to be computed for each classifier C_k and each class label θ_i. Let θ_j be the output label of a query instance z through a classifier C_k. The proximity measure d_j^k = $\phi(\theta_j, R_i^k)$ between the reference vector R_i^k and θ_j allows to estimate a basic belief assignment for each class and for each classifier. Given a classifier C_k, the proximity measure d_j^k relative to a class label θ_j is transformed into mass functions as follows:

$$m_j^k(\{\theta_j\}) = d_j^k, \qquad m_j^k(\Theta) = 1 - d_j^k \tag{2.46}$$

$$m_{\bar{j}}^k(\bar{\theta}_j) = 1 - \prod_{l \neq j} 1 - d_l^k, \qquad m_{\bar{j}}^k(\Theta) = \prod_{l \neq j} 1 - d_l^k \tag{2.47}$$

The belief associated to the classifier C_k and the label class θ_j is computed as the orthogonal combination of knowledge concerning θ_j (i.e. $m_j^k \oplus m_{\bar{j}}^k$). The evidence yielded by all the classifiers will then be merged through the orthogonal sum to get a confidence measure for each class label.

Rogova's work is seen as an extension of the idea proposed in (Mandler & Shurmann, 1988) when taking into consideration the process of the reference vectors computation. On the one hand, she has introduced a generic form of proximity measure enabling the use of different distance metrics when computing the class-conditional probabilities. On the other hand, she has attributed the greatest support to the label class θ_j by combining the masses m_j^k and $m_{\bar{j}}^k$ for a given classifier C_k.

2.3.4 Al-Ani's method

Al-Ani and Deriche (2002) have carried out a formalism for combining neural networks outputs within the the belief function framework. This method also calculates the piece of evidences on the basis of a distance metric between a classifier output and a reference vector. It begins by initializing the reference vectors for each class. Then, based on the training instances, it tries to optimize those vectors by minimizing the mean squared errors between the merged classifier outputs and the target outputs. Subsequently, the relation (the distance measure) between the classifier outputs and the optimized reference vectors will be defined as a pieces of evidence. Then, the combination method has to be done in the same manner as Rogova's method. Since the optimized reference vector is performing well compared with the static combination scheme, it entails additional training costs.

2.3.5 The calibration method

P. Xu, Davoine, and Denœux (2014) have suggested an evidential calibration method that converts the output of the SVM classifiers into evidence. This latter approach is then applied to the calibration and the combination of several SVM classifiers. Suppose we have a binary classification problem. Assume that $X = \{x_1, \ldots, x_N\}$ be a given training data with N objects that are characterized by class labels $y_j \in \Theta = \{0,1\}$. Let $s_j \in \mathbb{R}$ be the score yielded following to a pre-trained classifier of the i^{th} training instance having with class label y_j. The calibration process consists of estimating the posterior class probability $p(y = 1|s)$ of a given query instance with a score $s \in \mathbb{R}$ and an unknown class label. Although several calibration methods have been introduced in the literature, it has been proven that the logistic regression approach is the best suited for the calibration of maximum margin methods, more particularly the SVM one (Niculescu-Mizil & Caruana, 2005). Thus, the evidential logistic regression for binary SVM classifier calibration have been already introduced (P. Xu et al., 2014).

2.4 Conclusion

In this Chapter, we have discussed the belief function theory as an efficient tool within which the prediction of standard classifiers are merged for improving accuracy. In the beginning of this Chapter, we have pointed out the fundamental concepts of the belief function theory including knowledge representation, knowledge combination, etc. Then, we have presented some popular existing works for classifier ensemble within this theory.

It is substantial to note that the uncertainty may not be restricted to the classifier outputs. However, in several real world data, the attribute values may suffer from several aspects of uncertainty, including incompleteness and inconsistency. Since the belief function theory is a valuable tool for representing and managing all kinds of uncertainty, evidential data (i.e. meaning data with uncertain attribute values expressed within the belief function theory) have been introduced for several years yet (Lee, 1992). Despite their seriousness, classification problems from evidential data have not received the great attention till now. Our ultimate goal throughout the contribution part is to construct a classifier ensemble for addressing evidential data. One of the most important scientific challenges concerns the lack of classifiers processing such a kind of data. Thus, in the next Chapter, we develop three evidential classifiers. More particularly, we propose two decision tree classification versions that differ in the way of selecting decision attributes and a *k*-NN algorithm for addressing data with evidential attributes.

Chapter 3

New evidential classifiers

Contents

3.1 Introduction **39**
3.2 Decision tree classifiers for evidential data **40**
  3.2.1 Standard decision trees 40
  3.2.2 The structure of evidential training data 45
  3.2.3 Decision tree parameters 46
  3.2.4 Decision tree procedures 57
3.3 Enhanced Evidential *k*-Nearest Neighbors (EE*k*-NN) **65**
3.4 Comparative study **68**
  3.4.1 Experimentation settings 68
  3.4.2 Results discussions 70
3.5 Conclusion **74**

3.1 Introduction

Data uncertainty arises in several real world domains, including machine learning and pattern recognition applications. In classification problems, we could very well wind up with uncertain attribute values that are caused by sensor failures, measurements approximations or even subjective expert assessments, etc. To cope with uncertainty that pervades the attribute values, evidential databases have already been introduced (Lee, 1992). Despite their seriousness, there is a lack of machine learning algorithms processing such a kind of data. In this Chapter, we extend some well-known decision tree classifiers and the standard *k*-NN algorithm to an evidential context.

The remaining of this Chapter is organized as follows. Firstly, we describe standard decision tree classifiers. Then, we present our decision trees as well as our *k*-NN algorithm to process evidential data. A comparison between these three algorithms will be carried out on the basis of some assessment criteria.

3.2 Decision tree classifiers for evidential data

In this Section, we present firstly the principale of decision trees. Then we focus on decision trees within an evidential context. Thus, we present the structure of evidential data. We outline then on the parameters allowing the construction of our proposed decision tree classifiers. Subsequently, we highlight our decision tree procedures which mainly include the construction and the classification levels.

3.2.1 Standard decision trees

Decision trees are recognized among the most effective and efficient machine learning approaches and they have been successfully applied to solve real world problems within the artificial intelligence field. This success is mainly due to their great ability for solving complex problems through human-readable and computer-readable graphical representations. A plethora of algorithms have been introduced to construct decision trees from a given training set and to ensure the classification of query instances (Breiman, Friedman, Olshen, & Stone, 1984; Quinlan, 1986, 2014). The most used algorithms follow a Top Down Induction of Decision Tree approach (TDIDT) that consists on a recursive divide and conquer strategy by following the steps below:

- select, through the use of an attribute selection measure, the attribute that enables the best possible partitioning of the training set;

- split the current training data into training subsets according to the selected attribute values.
- nominate a training subset as a leaf when a stopping criterion is reached.

As regards the attribute selection process, several measures have been proposed in the literature (De Mántaras, 1991; Quinlan, 1986, 2014). The information gain, measuring the efficiency of an attribute when classifying the training instances, is one among the best known and most widely used measures. Given a training data S and an attribute A, the information gain will be set to:

$$Gain(S,A) = Info(S) - Info_A(S) \tag{3.1}$$

where

$$Info(S) = -\sum_{i=1}^{Q} p_i.log_2 p_i \tag{3.2}$$

and

$$Info_A(S) = -\sum_{v \in Domain(A)} \frac{|S_v^A|}{|S|} \tag{3.3}$$

where p_i reflects the proportion of objects having θ_i as class (i.e. $i \in \{1, \ldots, Q\}$) and S_v^A corresponds to the training subsets for which the attribute A has v as value.

One major limitation of this measure is that the attributes with the largest values are the most promoted ones (Quinlan, 2014). This had led to the introduction of the *GainRatio* measure used in the C4.5 algorithm (Quinlan, 1986, 2014). It is given as follows:

$$GainRatio(S,A) = \frac{Gain(S,A)}{SplitInfo(A)} \tag{3.4}$$

where

$$SpliInfo(A) = \sum_{v \in Domain(A)} \frac{|S_v^A|}{|S|}.log_2 \frac{|S_v^A|}{|S|}. \tag{3.5}$$

Example 3.1. *Let us treat the problem of the golf club managing. Sometimes staff are off duty and out of uniform, while the great majority of customers wish*

Table 3.1: An example of training data within a certain environment

	Outlook	Temperature	Humidity	Windy	Play
D_1	Sunny	Hot	High	False	No
D_2	Sunny	Hot	High	True	No
D_3	Overcast	Hot	High	False	Yes
D_4	Rain	Mild	High	False	Yes
D_5	Rain	Cool	Normal	True	No
D_6	Rain	Cool	Normal	False	No
D_7	Overcast	Cool	Normal	True	Yes
D_8	Sunny	Mild	High	False	No

to play golf (i.e. sometimes it is the opposite). Suppose that a manager has to optimize personnel availability, known that weather forecast may affect customer attendance. To meet the ultimate objective, the manager has to predict, on the basis of a knowledge base, the days when people want to play. Thus, he records approximately eight days of work D_j *with* $j \in \{1,2,3,4,5,6,7,8\}$. *Precisely, his uncertain initial knowledge about weather consists of four symbolic attributes:*

- ***Outlook*** *{Sunny, Overcast, rain}*
- ***Temperature*** *{Hold, Mild, Cool}*
- ***Humidity*** *{High, Normal}*
- ***Windy*** *{True, False}*

and a class label Play reflects customer attendance regarding the weather. This class label takes values in {Yes, No}, where ***Yes*** *implies the case where customers want to play in the day* D_j *and* ***No*** *refers to the case where customers do not want to play that day. The structure of the knowledge data is defined in Table 3.1.*

- *We compute firstly the information relative to the whole dataset as follows:*

$$Info(T) = -\frac{4}{8}log2\frac{4}{8} - \frac{4}{8}log2\frac{4}{8} = 1$$

- *We compute then the information after partitioning process with the Outlook, the Temperature, the Humidity and the Windy attributes:*

- ***Outlook:***

  $$Info_{Outlook} = \frac{3}{8} Info_{S^{Outlook}_{Sunny}} + \frac{2}{8} Info_{S^{Outlook}_{Overcast}} + \frac{3}{8} Info_{S^{Outlook}_{Rain}}$$

  where

  - $Info_{S^{Outlook}_{Sunny}} = -\frac{3}{3}\, log_2\, \frac{3}{3} = 0$
  - $Info_{S^{Outlook}_{Overcast}} = -\frac{2}{2}\, log_2\, \frac{2}{2} = 0$
  - $Info_{S^{Outlook}_{Rain}} = -\frac{1}{3}\, log_2\, \frac{1}{3} + -\frac{2}{3}\, log_2\, \frac{2}{3} = 0.92$

  then

  $$\begin{aligned} &Info_{Outlook} \\ &\qquad = 0.34 \end{aligned}$$

- ***Temperature:***

  $$Info_{Temperature} =$$

  where

  - $Info_{S^{Temperature}_{Hot}} = -\frac{2}{3}\, log_2\, \frac{2}{3} - \frac{1}{3}\, log_2\, \frac{1}{3} = 0.92$
  - $Info_{S^{Temperature}_{Mild}} = -\frac{1}{2}\, log_2\, \frac{1}{2} - \frac{1}{2}\, log_2\, \frac{1}{2} = 1$
  - $Info_{S^{Temperature}_{Cool}} = -\frac{2}{3}\, log_2\, \frac{2}{3} - \frac{1}{3}\, log_2\, \frac{1}{3} = 0.92$

  then

  $$\begin{aligned} Info_{Temperature} &= \frac{3}{8} * 0.92 + \frac{2}{8} * 1 + \frac{3}{8} * 0.92 \\ &= 0.94 \end{aligned}$$

- ***Humidity:***

  where

  - $Info_{S^{Humidity}_{High}} = -\frac{3}{5}\, log_2\, \frac{3}{5} - \frac{2}{5}\, log_2\, \frac{3}{5} = 0.97$
  - $Info_{S^{Humidity}_{Normal}} = -\frac{2}{3}\, log_2\, \frac{2}{3} - \frac{1}{3}\, log_2\, \frac{1}{3} = 0.92$

  then

  $$\begin{aligned} Info_{Humidity} &= \frac{5}{8} * 0.97 + \frac{3}{8} * 0.92 \\ &= 0.95 \end{aligned}$$

- ***Windy:***

  where

  - $Info_{S^{Windy}_{True}}= -\frac{2}{3}\ log_2\ \frac{2}{3}$ -$\frac{1}{3}\ log_2\ \frac{1}{3}$=*0.92*
  - $Info_{S^{Windy}_{False}}=-\frac{3}{5}\ log_2\ \frac{3}{5}$ -$\frac{2}{5}\ log_2\ \frac{2}{5}$=*0.97*

  then

$$\begin{aligned} Info_{Windy} &= \frac{3}{8} * 0.92 + \frac{5}{8} * 0.97 \\ &= 0.95 \end{aligned}$$

- *We compute the Gain for each attributes:*
  - *Gain(T,Outlook)=1-0.34=0.66*
  - *Gain(T,Temperature)=1-0.94=0.06*
  - *Gain(T,Humidity)=1-0.95=0.05*
  - *Gain(T,Windy)=1-0.95=0.05*

- *Before computing the Information Gain, we have to calculate firstly the split info for each attribute:*
  - *SplitInfo(T,Outlook)=*$-\frac{3}{8}log23_{8}$-$\frac{2}{8}log2\frac{2}{8}$-$\frac{3}{8}log2\frac{3}{8}$=*1.56*
  - *SplitInfo(T,Temperature)=*-$\frac{3}{8}log23_{8}$-$\frac{2}{8}log2\frac{2}{8}$-$\frac{3}{8}log2\frac{3}{8}$=*1.56*
  - *SplitInfo(T,Humidity)=*-$\frac{5}{8}log2\frac{5}{8}$-$\frac{3}{8}log2\frac{3}{8}$=*0.95*
  - *SplitInfo(T,Windy)=*-$\frac{5}{8}log2\frac{5}{8}$-$\frac{3}{8}log2\frac{3}{8}$=*0.95*

- *Finally, we compute the information Gain for each attribute:*
  - ***GainRatio(T,Outlook)=*** $\frac{0.66}{1.56}$***= 0.42***
  - *GainRatio(T,Temperature)=* $\frac{0.06}{1.56}$=*0.03*
  - *GainRatio(T,Humidity)=* $\frac{0.05}{0.95}$= *0.05*
  - *GainRatio(T,Windy)=* $\frac{0.05}{0.95}$=*0.05*

  Accordingly, the attribute Outlook will correspond to the root note of the three. In fact, it has the highest GainRatio.

In several real world applications, data may be uncertain due to some factors such as data randomness, data incompleteness, etc. However, classical decision tree versions did not possess the ability to adapt this kind of data. This shortcoming has led to the introduction of decision tree approaches to process the uncertain pervading the class labels. Examples include the fuzzy decision trees (Umano et al., 1994), the possibilistic decision trees (Hüllermeier, 2002; Jenhani et al., 2005, 2008), the uncertain decision trees (Qin, Xia, & Li, 2009) and the probabilistic decision trees (Quinlan, 1987). Although the probability theory is widely used for modeling uncertainty, several researchers have proved that probability cannot always be the adequate tool for representing data uncertainty, concretely the epistemic one (Nutter, 1986). Alternatively, the belief function theory has the advantage to represent all kinds of knowledge availability (Denœux, 1999). The process of incorporating the belief function theory into decision tree classifiers has been already developed (Elouedi et al., 2001; Trabelsi, Elouedi, & Mellouli, 2007; Vannoorenberghe & Denœux, 2002). To the best of our knowledge, almost all existing extended decision tree approaches within the belief function framework handle only the case of uncertain class label. Thus, by the following, we present the structure of evidential training data and we propose new decision tree classifier versions to address evidential data.

3.2.2 The structure of evidential training data

Let us remind that any classification problems require the construction of a learning algorithm from a given training data. This latter will be composed with instances (e.g. objects, persons, etc), where each one is described by a pair <Attributes, Class>. Within an uncertain context, an evidential database is composed by N objects x_j (i.e. $j \in \{1,\ldots,N\}$) where each of them is described by n attribute values $A = \{A_1,\ldots,A_n\}$ that are expressed within the belief function framework and a certain class label $y_j \in \Theta = \{\theta_1,\ldots,\theta_c\}$ (i.e c is the number of all possible class labels). Each attribute A_k (i.e. $k \in \{1,\ldots,n\}$) has a domain of discrete values denoted by Θ^{A_k}. An example of evidential training data is given below. Let us remind that any classification problems require the construction of a learning algorithm from a given training data. This latter will be composed with instances (e.g. objects, persons, etc), where each one is described by a pair <Attributes, Class>. Within an uncertain context, an evidential database is composed by N objects x_j (i.e. $j \in \{1,\ldots,N\}$) where each of them is described by n attribute values $A = \{A_1,\ldots,A_n\}$ that are expressed within the belief function framework and a certain class label $y_j \in \Theta = \{\theta_1, \dots, \theta_c\}$ (i.e c is the number of all possible class labels). Each attribute A_k (i.e. $k \in \{1, \dots, n\}$) has a domain of discrete values denoted by Θ^{A_k}. An example of evidential training data is given below.

Example 3.2. *Let us continue with Example 3.1 and suppose that the initial knowledge about weather are known with uncertainty that are expressed within the belief function framework. The structure of the knowledge data is defined in Table 3.2.*

Table 3.2: Evidential training data

	$Outlook$	$Temperature$	$Humidity$	$Wind$	$Play$
D_1	$m_1^{\Theta^{Outlook}}(\{Sunny\})$=1	$m_1^{\Theta^{Temperature}}(\{Hot\})$=0.75 $m_1^{\Theta^{Temperature}}(\{Mild\})$=0.25	$m_1^{\Theta^{Humidity}}(\{High\})$=0.65 $m_1^{\Theta^{Humidity}}(\{Normal\})$=0.35	$m_1^{\Theta^{Windy}}(\{True\})$=1	Yes
D_2	$m_2^{\Theta^{Outlook}}(\{Sunny\})$=0.8 $m_2^{\Theta^{Outlook}}(\{Overcast\})$=0.2	$m_2^{\Theta^{Temperature}}(\{Hot\})$=0.65 $m_2^{\Theta^{Temperature}}(\{Mild\})$=0.35	$m_2^{\Theta^{Humidity}}(\{High\})$=0.65 $m_2^{\Theta^{Humidity}}(\{Normal\})$=0.20 $m_2^{\Theta^{Humidity}}(\Theta^{Humidity})$=0.15	$m_2^{\Theta^{Windy}}(\{False\})$=1	Yes
D_3	$m_3^{\Theta^{Outlook}}(\{Overcast\})$=1	$m_3^{\Theta^{Temperature}}(\{Hot\})$=1	$m_3^{\Theta^{Humidity}}(\{High\})$=0.50 $m_3^{\Theta^{Humidity}}(\Theta^{Humidity})$=0.50	$m_3^{\Theta^{Windy}}(\{False\})$=0.80 $m_3^{\Theta^{Windy}}(\Theta^{Windy})$=0.20	No
D_4	$m_4^{\Theta^{Outlook}}(\{Rain\})$=0.85 $m_4^{\Theta^{Outlook}}(\{Overcast\})$=0.15	$m_4^{\Theta^{Temperature}}(\{Mild\})$=1	$m_4^{\Theta^{Humidity}}(\{High\})$=0.95 $m_4^{\Theta^{Humidity}}(\{Normal\})$=0.05	$m_4^{\Theta^{Windy}}(\{False\})$=1	No
D_5	$m_5^{\Theta^{Outlook}}(\{Rain\})$=1	$m_5^{\Theta^{Temperature}}(\{Cool\})$=0.63 $m_5^{\Theta^{Temperature}}(\Theta^{Temperature})$=0.37	$m_5^{\Theta^{Humidity}}(\{High\})$=1	$m_5^{\Theta^{Windy}}(\{False\})$=0.89 $m_5^{\Theta^{Windy}}(\{True\})$=0.11	No
D_6	$m_6^{\Theta^{Outlook}}(\{Overcast\})$=0.60 $m_6^{\Theta^{Outlook}}(\{Rain\})$=0.40	$m_6^{\Theta^{Temperature}}(\{Cool\})$=1	$m_6^{\Theta^{Humidity}}(\{Normal\})$=0.75 $m_6^{\Theta^{Humidity}}(\{High\})$=0.25	$m_6^{\Theta^{Windy}}(\{True\})$=1	Yes
D_7	$m_7^{\Theta^{Outlook}}(\{Sunny\})$=1	$m_7^{\Theta^{Temperature}}(\{Hot\})$=0.55 $m_7^{\Theta^{Temperature}}(\{Mild\})$=0.45	$m_7^{\Theta^{Humidity}}(\{High\})$=0.74 $m_7^{\Theta^{Humidity}}(\Theta^{Humidity})$=0.26	$m_7^{\Theta^{Windy}}(\{False\})$=0.95 $m_7^{\Theta^{Windy}}(\{True\})$=0.05	No
D_8	$m_8^{\Theta^{Outlook}}(\{Sunny\})$=1	$m_8^{\Theta^{Temperature}}(\{Cool\})$=0.90 $m_8^{\Theta^{Temperature}}(\{Mild\})$=0.10	$m_8^{\Theta^{Humidity}}(\{Normal\})$=0.88 $m_8^{\Theta^{Humidity}}(\{High\})$=0.12	$m_8^{\Theta^{Windy}}(\{False\})$=1	Yes

3.2.3 Decision tree parameters

This subsection is devoted to highlighting the main parameters enabling the construction of decision tree classifiers from data with evidential attributes. Four main parameters conduct to the construction of our proposed approaches. We provide a detailed description for each parameter when relying on following notations:

- T: a given training set composed by N objects x_j; $j = \{1,\ldots,N\}$.

- S: a subset of objects belonging to the training set T.

- A=$\{A_1,\ldots,A_n\}$: the set of n attributes.

- Θ^{A_k}: corresponds to all the possible values of an attribute $A_k \in A$ where $k = \{1,\ldots,n\}$.

- Θ= $\{\theta_1,\ldots,\theta_c\}$: represents the c possible classes of the classification problem.

- $S_v^{A_k}$: For each value $v \in \Theta^{A_k}$, we define the subset $S_v^{A_k}$ composed with objects having v as a value.

- $m_j^{\Theta^{A_k}}(v)$: denotes the bbm assigned to the hypothesis that the actual attribute value of object x_j belongs to $v \subseteq \Theta^{A_k}$.

- $m^{\Theta^{A_v^k}}$: is the certain bba corresponding to the attribute A^k and having v as its unique focal element.

- m_j^{Θ}: corresponds to the bba relative to the class of the object x_j.

- L= $\{L_1,\ldots,L_F\}$: represents the F generated leaves when building the decision tree.

Attribute selection measure

The attribute selection measure is considered as one of the major parameters ensuring decision tree construction. It consists of choosing, for each decision node of the tree, the attribute test that will best separate the training instances into homogenous subsets. The *GainRatio* and the *DiffRatio* are two commonly attribute selection measures. The former one is building primarily upon the work conducted by Quinlan (1986), while the latter one is mainly relied on the intra distance between training instances within the TBM framework (Elouedi et al., 2001). Accordingly, we suggest to construct two decision tree classifiers, on the basis of the *GainRatio* and the *DiffRatio* measures, to process data with evidential attributes.

The *GainRatio* measure: This measure is relied on the entropy calculated from the average probability obtained from the set of objects in the node. With the aim of choosing the most appropriate attribute, we propose the following steps:

1. We compute the average probability $Pr\{S\}(\theta_i)$ relative to each class by taking into account the set of objects S:

   $$Pr\{S\}(\theta_i) = \frac{1}{\sum_{x_j \in S} P_j^S} \sum_{x_j \in S} P_j^S \gamma_{ij} \tag{3.6}$$

   where γ_{ij} equals 1 if the object x_j belongs to the class θ_i, 0 otherwise and P_j^S corresponds to the probability of the object x_j to belong to the subset S. Assume the independence between attributes, the probability P_j^S will be equal to the product of the different pignistic probabilities induced from the attribute bbas of the object x_j and enabling x_j to belong to the node S. Let $A_B = \{A_1, \ldots, A_O\} \in A$ with values $V_B = \{v_1, \ldots, v_O\}$ be the set of attributes leading to the branch S, the probability P_j^S will be set to:

   $$P_j^S = \prod_{A_o \in A_B} BetP^{\Theta^{A_o}}[x_j](v_o) \tag{3.7}$$

2. We compute the entropy $Info(S)$ of the average probabilities in S:

   $$Info(S) = -\sum_{i=1}^{q} Pr\{S\}(\theta_i) log_2 Pr\{S\}(\theta_i) \tag{3.8}$$

3. Considering an attribute A_k, for each value $v \in \Theta^{A_k}$, we define the subset $S_v^{A_k}$ with objects having v as attribute value. As we handle attribute values, the subset $S_v^{A_k}$ will contain objects x_j such that their attribute pignistic probabilities of v is computed satisfying:

   $$BetP^{\Theta^{A_k}}\{x_j\}(v) \neq 0 \tag{3.9}$$

4. We compute, for objects in subset $S_v^{A_k}$, the average probability $Pr\{S_v^{A_k}(\theta_i\}$ of the class θ_i (i.e. $v \in \Theta^{A_k}$ and $A_k \in A$):

   $$Pr\{S_v^{A_k}\}(\theta_i) = \frac{1}{\sum_{x_j \in S_v^{A_k}} P_j^{S_v^{A_k}}} \sum_{x_j \in S_v^{A_k}} P_j^{S_v^{A_k}} \gamma_{ij} \tag{3.10}$$

where $P_j^{S_v^{A_k}}$ is the probability of the object x_j to belong to the subset $S_v^{A_k}$ (its computation is done in the same manner as the computation of P_j^S).

5. We compute $Info_{A_k}(S)$ as discussed in (Quinlan, 1986), while using the probability distribution instead of the proportions. Assume $|S| = \sum_{x_j \in S} P_j^S$ and $|S_v^{A_k}| = \sum_{x_j \in S_v^{A_k}} P_j^{S_v^{A_k}}$, we get:

$$Info_{A_k}(S) = \sum_{v \in \Theta^{A_k}} \frac{\sum_{x_j \in S_v^{A_k}} P_j^{S_v^{A_k}}}{|S|} Info(S_v^{A_k}) \tag{3.11}$$

   where $Info(S_v^{A_k})$ is calculated from Equation 3.8.

6. We compute the information gain yielded by the attribute A_k over the set of objects S such that:

$$Gain(S, A_k) = Info(S) - Info_{A_k}(S) \tag{3.12}$$

7. We compute the $GainRatio$ relative to the attribute A_k by the use of the $SplitInfo$

$$GainRatio(S, A_k) = \frac{Gain(S, A_k)}{SplitInfo(S, A_k)} \tag{3.13}$$

   where the $SplitInfo$ value is defined as follows:

$$SplitInfo(S, A_k) = - \sum_{v \in \Theta^{A_k}} \frac{\sum_{x_j \in S_v^{A_k}} P_j^{S_v^{A_k}}}{|S|} log_2 \frac{\sum_{x_j \in S_v^{A_k}} P_j^{S_v^{A_k}}}{|S|} \tag{3.14}$$

8. We repeat the same process for each attribute $A_k \in A$ (from step 3 to step 7) and then we select the one that has the maximum $GainRatio$.

Example 3.3. *Let us consider the evidential training data given in Example 3.1. We try to illustrate the attribute selection process, when relied on the GainRatio measure:*

- *We start by computing the entropy $Info(S)$ using Equation 3.8*

$$\begin{aligned} Info(S) &= -Pr\{S\}(Yes)log_2Pr\{S\}(Yes) - Pr\{S\}(No)log_2Pr\{S\}(No) \\ &= -\frac{4}{8}log_2\frac{4}{8} - \frac{4}{8}log_2\frac{4}{8} \\ &= 1 \end{aligned}$$

- *To determine the information gain of the attributes Outlook, Temperature, Humidity and Windy, we have firstly to compute the probability of the belonging of an object (D_j) to the subsets $S^{Outlook}_{Sunny}$, $S^{Outlook}_{Overcast}$, $S^{Outlook}_{Rain}$, $S^{Temperature}_{Hot}$, $S^{Temperature}_{Mild}$, $S^{Temperature}_{Cool}$, $S^{Humidity}_{High}$, $S^{Humedity}_{Normal}$, S^{Windy}_{True} and S^{Windy}_{False}. The probability of belonging of objects in terms of the attribute values are given in Table 3.3*

Table 3.3: The probability of belonging of objects in terms of the attribute values.

	Outlook		
	$P^{S^{Outlook}_{Sunny}}$	$P^{S^{Outlook}_{Overcast}}$	$P^{S^{Outlook}_{Rain}}$
D_1	1	0	0
D_2	0.8	0.2	0
D_3	0	1	0
D_4	0	0.15	0.85
D_5	0	0	1
D_6	0	0.6	0.4
D_7	1	0	0
D_8	1	0	0

	Temperature		
	$P^{S^{Temperature}_{Hot}}$	$P^{S^{Temperature}_{Mild}}$	$P^{S^{Temperature}_{Cool}}$
D_1	0.75	0.25	0
D_2	0.65	0.35	0
D_3	1	0	0
D_4	0	1	0
D_5	0.12	0.12	0.76
D_6	0	0	1
D_7	0.55	0.45	0
D_8	0	0.1	0.9

	Humidity	
	$P^{S^{Humidity}_{High}}$	$P^{S^{Humidity}_{Normal}}$
D_1	0.65	0.35
D_2	0.73	0.27
D_3	0.75	0.25
D_4	0.95	0.05
D_5	1	0
D_6	0.25	0.75
D_7	0.87	0.13
D_8	0.12	0.88

	Windy	
	$P^{S^{Windy}_{True}}$	$P^{S^{Windy}_{False}}$
D_1	1	0
D_2	0	1
D_3	0.1	0.9
D_4	0	1
D_5	0.11	0.89
D_6	1	0
D_7	0.05	0.95
D_8	0	1

- *We move on now to compute the average probability associated with each class according to Equation 3.10. The results are given from Table 3.4 to Table 3.7.*

Table 3.4: Average Probability associated to the attribute Outlook

	Sunny	Overcast	Rain
Yes	0.74	0.41	0.18
No	0.26	0.59	0.82

Table 3.5: Average Probability associated to the attribute Temperature

	Hot	Mild	Cool
Yes	0.46	0.31	0.72
No	0.54	0.69	0.28

Table 3.6: Average Probability associated to the attribute Humidity

	High	Normal
Yes	0.33	0.82
No	0.67	0.18

Table 3.7: Average Probability associated to the attribute Windy

	True	False
Yes	0.89	0.35
No	0.11	0.65

- *The next step consists of calculating the information relative to the four uncertain attributes (see Equation 3.11):*
  - ***Outlook:***

$$Info_{Outlook} = \frac{3.8}{8} Info_{S^{Outlook}_{Sunny}} + \frac{1.95}{8} Info_{S^{Outlook}_{Overcast}} + \frac{2.25}{8} Info_{S^{Outlook}_{Rain}}$$

    where

    - $Info_{S^{Outlook}_{Sunny}}$ = -0.74 log_2 0.74 - 0.26 log_2 0.26=0.83
    - $Info_{S^{Outlook}_{Overcast}}$ = -0.41 log_2 0.41 - 0.59 log_2 0.59=0.97
    - $Info_{S^{Outlook}_{Rain}}$ = -0.17 log_2 0.17 - 0.82 log_2 0.82=0.68

then

$$
\begin{aligned}
Info_{Outlook} &= \frac{3.8}{8} * 0.83 + \frac{1.95}{8} * 0.97 + \frac{2.25}{8} * 0.68 \\
&= 0.82
\end{aligned}
$$

- ***Temperature:***

$$
\begin{aligned}
Info_{Temperature} =& \frac{3.07}{8} Info_{S^{Temperature}_{Hot}} + \qquad \frac{2.27}{8} Info_{S^{Temperature}_{Mild}} \\
&+ \frac{2.66}{8} Info_{S^{Temperature}_{Cool}}
\end{aligned}
$$

where

  - $Info_{S^{Temperature}_{Hot}} = -0.46 \log_2 0.46 - 0.54 \log_2 0.54 = 0.99$
  - $Info_{S^{Temperature}_{Mild}} = -0.31 \log_2 0.31 - 0.69 \log_2 0.69 = 0.89$
  - $Info_{S^{Temperature}_{Cool}} = -0.72 \log_2 0.72 - 0.28 \log_2 0.28 = 0.86$

then

$$
\begin{aligned}
Info_{Temperature} &= \frac{3.07}{8} * 0.99 + \frac{2.27}{8} * 0.89 + \frac{2.65}{8} * 0.86 \\
&= 0.92
\end{aligned}
$$

- ***Humidity:***

$$
Info_{Humidity} = \frac{5.32}{8} Info_{S^{Humidity}_{High}} + \frac{2.68}{8} Info_{S^{Humidity}_{Normal}}
$$

where

  - $Info_{S^{Humidity}_{High}} = -0.33 \log_2 0.33 - 0.67 \log_2 0.67 = 0.91$
  - $Info_{S^{Humidity}_{Normal}} = -0.82 \log_2 0.82 - 0.18 \log_2 0.18 = 0.68$

then

$$
\begin{aligned}
Info_{Humidity} &= \frac{5.32}{8} * 0.91 + \frac{2.68}{8} * 0.68 \\
&= 0.83
\end{aligned}
$$

- ***Windy:***

$$
Info_{Windy} = \frac{2.26}{8} Info_{S^{Windy}_{True}} + \frac{5.74}{8} Info_{S^{Humidity}_{False}}
$$

where

- $* \; Info_{S^{Windy}_{True}}$= -0.89 log_2 0.89 - 0.11 log_2 0.11=0.51
- $* \; Info_{S^{Windy}_{False}}$= -0.35 log_2 0.35 - 0.65 log_2 0.65=0.93

then

$$\begin{aligned} Info_{Windy} &= \frac{2.26}{8} * 0.51 + \frac{5.74}{8} * 0.93 \\ &= 0.81 \end{aligned}$$

- *Let us move on now to compute the gain obtained by each attribute over the set of objects (see Equation 3.12):*
  - *Gain(S, Outlook)=1-0.83=0.17*
  - *Gain(S, Temperature)=1-0.92=0.08*
  - *Gain(S, Humidity)=1-0.82=0.18*
  - *Gain(S, Windy)=1-0.81=0.19*
- *Subsequently, we compute the $SplitInfo$ corresponds to each attribute:*
  - *SplitInfo(S, Outlook)=* $-\frac{3.8}{8} \; log_2 \; \frac{3.8}{8} - \frac{1.95}{8} \; log_2 \; \frac{1.95}{8} - \frac{2.25}{8} \; log_2 \; \frac{2.25}{8} = 1.52$
  - *SplitInfo(S, Temperature)=* $-\frac{3.07}{8} \; log_2 \; \frac{3.07}{8} - \frac{2.27}{8} \; log_2 \; \frac{2.27}{8} - \frac{2.65}{8} \; log_2 \; \frac{2.65}{8} = 1.57$
  - *SplitInfo(S, Humidity)=* $-\frac{5.31}{8} \; log_2 \; \frac{5.31}{8} - \frac{2.68}{8} \; log_2 \; \frac{2.68}{8} = 0.92$
  - *SplitInfo(S, Windy)=*$-\frac{2.26}{8} \; log_2 \; \frac{2.26}{8} - \frac{5.74}{8} \; log_2 \; \frac{5.74}{8} = 0.85$
- *Finally, we compute the GainRatio relative to every attribute:*
  - *GainRatio(S, Outlook)=* $\frac{0.18}{1.52} = 0.12$
  - *GainRatio(S, Temperature)=* $\frac{0.08}{1.57} = 0.05$
  - *GainRatio(S, Humidity)=* $\frac{0.17}{0.92} = 0.18$
  - ***GainRatio(S, Windy)=*** $\frac{0.19}{0.85}$***=0.22***

Accordingly, we can deduce that the Windy attribute has the highest $GainRatio$. Thus, it will be considered as the root node of the tree.

The *DiffRatio* measure: This measure consists of computing the intra-group distance that measures for each attribute value how much objects are close to each other. We propose the following steps to pick out the best attribute:

1. We compute the total distance taken over the training set T as follows:
$$SumD(S) = \sum_{x_i \in S} \sum_{x_{j \geq i+1} \in S} P_i^S . P_j^S \lambda_{i,j} \tag{3.15}$$
where $\lambda_{i,j}$ equals 1 if both objects x_i and x_j have the same class label and P_i^S states the probability of belonging of the object x_i to the set S. It is calculated as the cross product of the pignistic probabilities of the different attribute bbas relative to an object x_i and allowing x_i to belong to S.
2. Then, for each attribute value v, we compute $SumD(S_v^{A_k})$ as follows:
$$SumD(S_v^{A_k}) = \sum_{x_i \in S_v^{A_k}} \sum_{x_{j \geq i+1} \in S_v^{A_k}} P_i^{S_v^{A_k}} . P_j^{S_v^{A_k}} \lambda_{i,j} \tag{3.16}$$
where $P_i^{S_v^{A_k}}$ quantifies the probability of the object x_i to belong to the subset $S_v^{A_k}$. Note that it is computed as the same manner as the computation of P_j^s.
3. Once the different $SumD(S_v^{A_k})$ are calculated, for each attribute $A_k \in A$, we compute $SumD_{A_k}(S)$ as follows:
$$SumD_{A_k}(S) = \sum_{v \in \Theta^{A_k}} SumD(S_v^{A_k}) \tag{3.17}$$
4. In analogy to classical decision trees, we compute the difference before and after performing the partition process by the attribute A_k. This measure, denoted by $diff(S,A_k)$, is defined as the difference between $SumD(S)$ and $SumD_{A_k}(S)$ as follows:
$$diff(S,A_k) = SumD(S) - SumD_{A_k}(S) \tag{3.18}$$
5. Using the $SplitInfo$, we compute the $DiffRatio$ relative to the attribute A_k.
$$DiffRatio(S,A_k) = \frac{diff(S,A_k)}{SplitInfo(S,A_k)} \tag{3.19}$$
where
$$SplitInfo(S,A_k) = -\sum_{v \in D(A_k)} \frac{\sum_{x_j \in S_v^{A_k}} P_j^{S_v^{A_k}}}{|S|} log_2 \frac{\sum_{x_j \in S_v^{A_k}} P_j^{S_v^{A_k}}}{|S|} \tag{3.20}$$

6. We repeat this process for each attribute $A_k \in A$ and then select the one that maximize the $DiffRatio$.

Example 3.4. *Suppose that we have an evidential training data as presented in Example 3.1. Let us try to illustrate the* $DiffRatio$ *attribute selection measure.*

- *Let us start by computing the distance* $SumD(S)$ *which is defined as the sum of distances between each training instance and the whole set S as follows:*

$$\begin{aligned} SumD(S) &= \sum_{D_i \in S} \sum_{D_{j \geq i+1} \in S} P_i^S . P_j^S \lambda_{i,j} \\ &= 12 \end{aligned}$$

- *Subsequently, we compute the distance for each attribute values:*
  - ***Outlook:***
    - $SumD(S_{Sunny}^{Outlook})= 1$
    - $SumD(S_{Overcast}^{Outlook})=0.15$
    - $SumD(S_{Rain}^{Outlook})=0.85$
  - ***Temperature:***
    - $SumD(S_{Hot}^{Temperature})= 0.74$
    - $SumD(S_{Mild}^{Temperature})= 0.65$
    - $SumD(S_{Cool}^{Temperature})= 0.9$
  - ***Humidity:***
    - $SumD(S_{High}^{Humidity})= 5.03$
    - $SumD(S_{Normal}^{Humidity})= 1.28$
  - ***Windy:***
    - $SumD(S_{True}^{Windy})= 1.02$
    - $SumD(S_{False}^{Windy})= 5.24$
- *We compute,* $SumD_{Outlook}$, $SumD_{Temperature}$, $SumD_{Humidity}$, $SumD_{Windy}$ *for respectively the attributes Outlook, Temperature, Humidity and Windy:*
  - $SumD_{Outlook}=SumD(S_{Sunny}^{Outlook})+SumD(S_{Overcast}^{Outlook})+SumD(S_{Rain}^{Outlook})$ $=1+0.15+0.85=2$
  - $SumD_{Temperature}=SumD(S_{Hot}^{Temperature})+SumD(S_{Mild}^{Temperature})+SumD(S_{Cool}^{Temperature})$ $=0.74+0.65+0.9=2.29$

  - $SumD_{Humidity}$=$SumD(S^{Humidity}_{High})$+$SumD(S^{Humidity}_{Normal})$ =5.03+1.28=6.31
  - $SumD_{Windy}$=$SumD(S^{Windy}_{True})$+$SumD(S^{Windy}_{False})$ =1.02+5.24=6.26

- *As a next step, we compute the difference induced when partitioning the tree through the four attributes:*
  - $diff(S,Outlook)$= $SumD(S) - SumD_{Outlook}$= 12-2= 10
  - $diff(S,Temperature)$= $SumD(S) - SumD_{Temperature}$= 12-2.29=9.71
  - $diff(S,Humidity)$= $SumD(S) - SumD_{Humidity}$=12-6.31=5.69
  - $diff(S,Windy)$= $SumD(S) - SumD_{OWindy}$=12-6.26=5.74

- *Finally, we turn to compute the $DiffRatio$ relative to each attribute:*
  - $DiffRatio(S,Outlook)= \frac{diff(S,Outlook)}{SplitInfo(S,Outlook)} = \frac{10}{1.52} = 6.57$
  - $DiffRatio(S,Temperature)= \frac{diff(S,Temperature)}{SplitInfo(S,Temperature)} = \frac{9.71}{1.57} = 6.18$
  - $DiffRatio(S,Humidity)= \frac{diff(S,Humidity)}{SplitInfo(S,Humidity)} = \frac{5.69}{0.92} = 6.18$
  - $DiffRatio(S,Windy)= \frac{diff(S,Windy)}{SplitInfo(S,Windy)} = \frac{5.73}{0.85} = 6.74$

From the yielded results, we can remark that the Windy corresponds to the attribute with the greatest $DiffRatio$. Accordingly, this latter will be regarded as the root node of the decision tree.

Partitioning strategy

The splitting strategy consists of dividing the training set according to the values of the chosen attribute A_k, meaning that a branch will be associated to each value v of the chosen attribute and each edge will contain a subset $S_v^{A_k}$ from S. Since we handle data with evidential attributes, each training object may be part of more than one subset. That is, each training object may belong to more than one branch with a probability of belonging computed in terms of the pignistic probability. To put it simply, a given object x_j has to be assigned to each branch having v as value and satisfying $BetP^{\Theta^{A_k}}[x_j](v) \neq 0$.

Stopping criteria

The stopping criteria are quite similar to those used by the standard decision tree. There exist mainly four stopping strategies:

1. Only one instance is part to the treated node.
2. Instances of the treated node belong to the same class.
3. There is no further attribute for checking.
4. The remaining attributes have $GainRatio$ or $DiffRatio$ equal or less than zero.

Structure of leaves

Our ultimate purpose is to construct decision tree from data with evidential attributes. In such a case, an object x_i may belong to more than one leaf with a probability of belonging denoted by $P_i^{L_f}$. As leaves may include objects with different class values, our proposed decision tree building algorithm assigns for each leaf a probability distribution over the set of classes computed from the probability of objects belonging to this leaf. The probability distribution relative to L_f over a class $C_q \in C$ is set to:

$$Pr\{L_f\}(\theta_q) = \frac{1}{\sum_{x_i \in L_f} P_i^{L_f}} \sum_{x_i \in L_f} P_i^{L_f} \gamma_{iq} \tag{3.21}$$

where γ_{iq} equals 1 if the class of the object x_i is θ_q, 0 otherwise and $P_i^{L_f}$ is the probability of the instance x_i to belong to the leaf L_f. This latter is calculated as the cross product of the pignistic probabilities of the object x_i to belong to the nodes that link the root node and the corresponding leaf node L_f.

3.2.4 Decision tree procedures

Construction level

The construction of our proposed decision tree classifier within an uncertain environment follows Quinlan's algorithm steps. It relies on a top down construction approach. Assume that T is our learning set, the different steps of our decision tree algorithm will be set as follows:

1. We start by creating the root node from the whole learning set T.
2. We check if the root node satisfies any stopping criteria.
    - If one stopping criterion is reached, the treated node will be declared as a leaf for which we compute the probability distribution over the set of classes.
    - else, we pick out the attribute that maximizes the attribute selection measure presented previously. The chosen one will be the root node of our decision tree relative to the set T.
3. We create a branch for each attribute value chosen as a root. This partitioning step leads to several subsets where each one contains close objects according to the attribute value.
4. We restart the same process from level 2 until all nodes are considered as leaves.

Example 3.5. *Let us continue with the evidential training data given in Table 3.2. Our ultimate goal is to construct belief decision trees that are based on the GainRatio and $DiffRatio$ attribute selection criteria. As previously mentioned, we resort to both the GainRatio and the $DiffRatio$ criteria. Figure 3.1 and Figure 3.2 correspond respectively to the constructed trees with the GainRatio and the $DiffRatio$ criteria. The probability distribution of leaves for the GainRatio decision tree and the $DiffRatio$ decision trees are given respectively in Table 3.8 and Table 3.9.*

Table 3.8: Probability distribution of leaves according to the $GainRatio$ Criterion

	$Pr\{L_f\}(Yes)$	$Pr\{L_f\}(No)$
L_1	0.95	0.05
L_2	0	1
L_3	0	1
L_4	0.83	0.17
L_5	0.75	0.25
L_6	0.45	0.55
L_7	0.37	0.63
L_8	1	0
L_9	0.9	0.1
L_{10}	0.97	0.032
L_{11}	1	0
L_{12}	0.68	0.32
L_{13}	0.75	0.25
L_{14}	1	0
L_{15}	0.15	0.85
L_{16}	0.19	0.81
L_{17}	0	1

Table 3.9: Probability distribution of leaves according to the $DiffRatio$ Criterion

	$Pr\{L_f\}(Yes)$	$Pr\{L_f\}(No)$
L_1	0.95	0.05
L_2	0.86	0.14
L_3	0.78	0.22
L_4	0.45	0.55
L_5	0.37	0.63
L_6	1	0
L_7	0.67	0.32
L_8	0.75	0.25
L_9	1	0
L_{10}	0.13	0.87
L_{11}	0.32	0.68
L_{12}	0	1

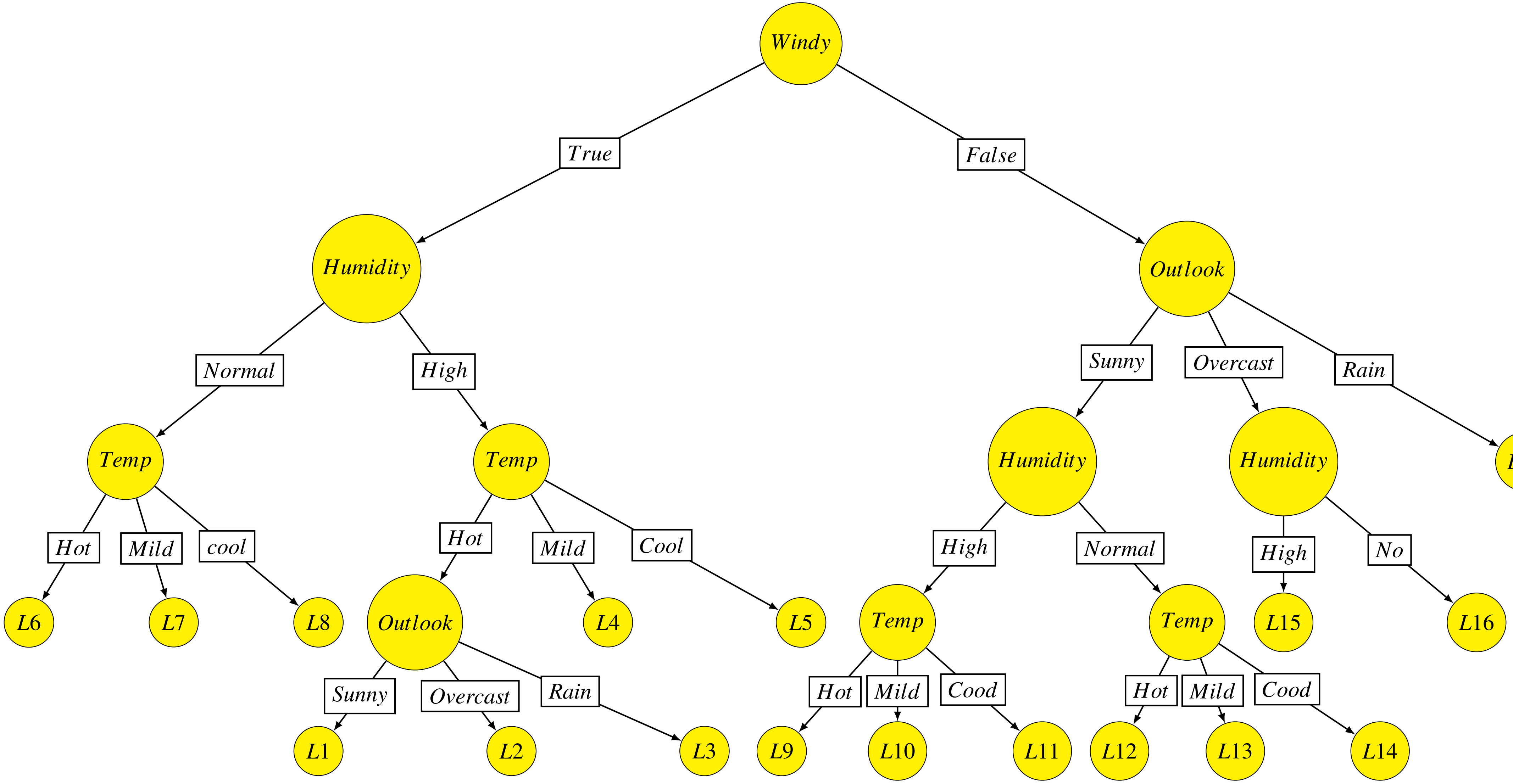


Figure 3.1: Decision tree according to the Gain Ratio Criterion

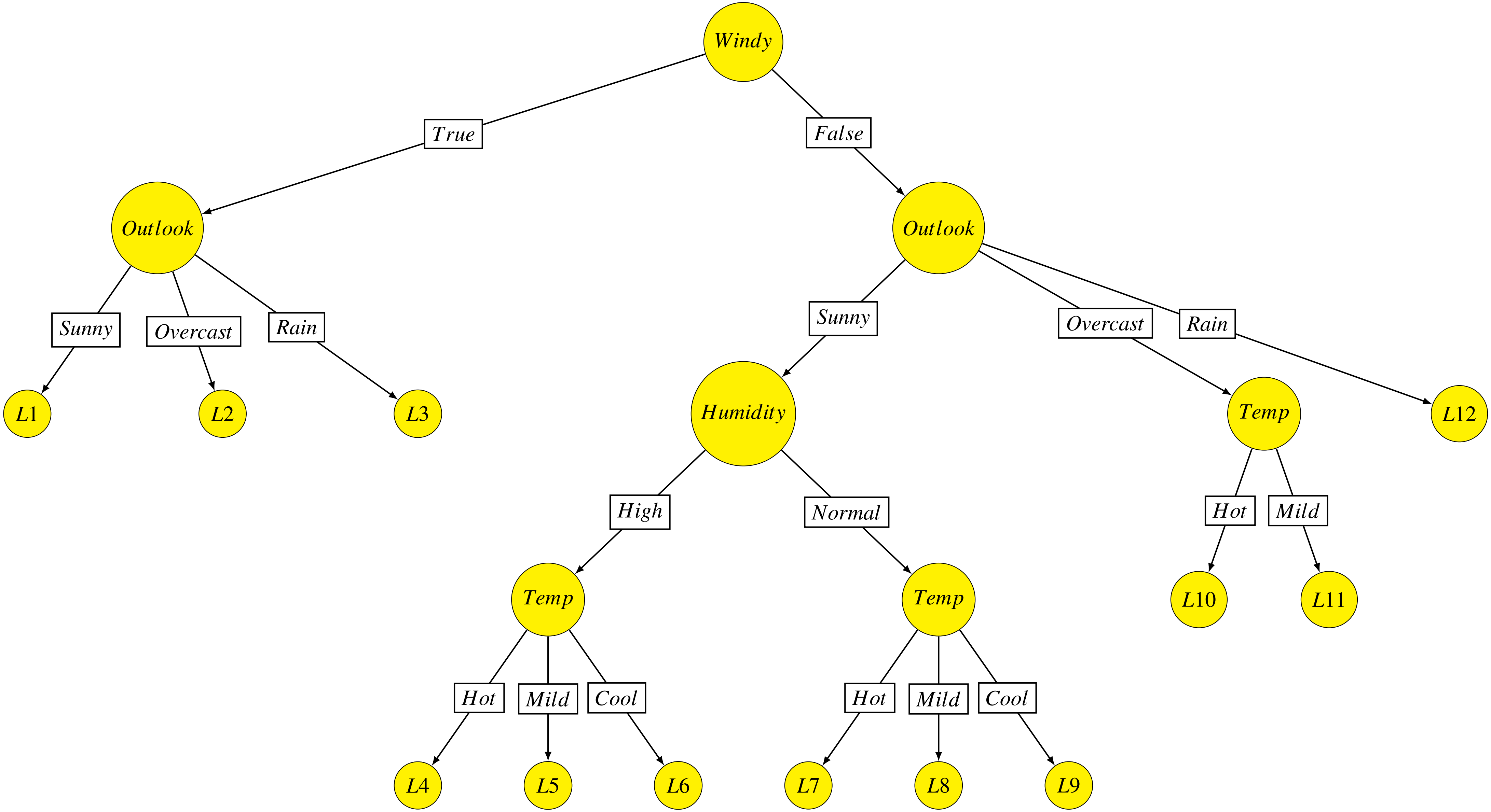


Figure 3.2: Decision tree according to the Diff Ratio Criterion

Classification level

As stated by Quinlan (1987), a decision tree paradigm consists mainly in two distinct procedures: the construction and the classification steps. Herein, we propose a novel approach for classifying objects with evidential attributes. Let z be a query instance described by a set of attributes $A = \{A_1, \ldots, A_n\}$. The global frame of discernment relative to all the attributes, denoted by Θ^A, is equal to the cross product of the different Θ^{A_k} as follows:

$$\Theta^A = \underset{k=1,\ldots,n}{\times} \Theta^{A_k}. \tag{3.22}$$

Since objects are described by a combination of values, we compute, for query instance z, the joint bba expressing beliefs on its attribute values. To do so, we proceed as follows:

- We extend the different bbas $m_z^{\Theta^{A_k}}$ to the global frame of discernment Θ^A (see Equation 3.22) for getting the different bbas $m_z^{\Theta^{A_k} \uparrow \Theta^A}$.

- We combine the different extended bbas using the conjunctive operator:

$$m_z^{\Theta^A} = \underset{k=1,\ldots,n}{\bigcirc\!\!\!\!\cap} \; m_z^{\Theta^{A_k} \uparrow \Theta^A} \tag{3.23}$$

Once the joint bba $m_z^{\Theta^A}$ is obtained, we move on to compute the probability distribution $Pr_z[x](\theta_q)$ of each focal element x. The computation of this probability distribution depends mainly on the focal elements of the bba m^{Θ^A} and on the subset x:

- When x is a singleton, the probability distribution $Pr_z[x](\theta_q)$ corresponds to the probability assigned to the class θ_q of the leaf that is attached to the focal element.

- else, we explore all possible paths correspond to this combination of values. There are two possible cases:
    - The case 1 is that all paths lead to the same leaf. In this case, the probability $Pr_z[x](\theta_q)$ will be equal to the probability of assigned to the class θ_q of the corresponding leaf.

  - The case 2 is that paths lead to distinct leaves. The probability $Pr_z[x](\theta_q)$ of the class θ_q corresponds to the average probability of the class θ_q relative to the different attached leaves.

- Finally, the probability distribution relative to each object test z over the set of classes will be set to:

$$Pr_z(\theta_q) = \sum_{x \subseteq \Theta^A} m^{\Theta^A}(x) Pr_z[x](\theta_q) \; \forall \, q \in \{1, \ldots, c\} \quad (3.24)$$

  The most probable class of the object z is the one with the highest probability distribution.

Example 3.6. *Suppose that the Golf's manager has beliefs concerning the weather of a day D_9 and he wants to estimate customer attendance. To do so, he has to classify the new instance through the decision tree constructed from the data given in Table 3.2. The weather's beliefs are defined in Table 3.10:*

Table 3.10: Evidential attribute values for a query instance

	Outlook	*Temperature*	*Humidity*	*Wind*	*Play*
D_9	$m_9^{\Theta^{Outlook}}(\{Sunny\})$=0.6 $m_9^{\Theta^{Outlook}}(\{Overcast\})$=0.4	$m_9^{\Theta^{Temperature}}(\{Hot\})$=0.5 $m_9^{\Theta^{Temperature}}(\{Mild\})$=0.5	$m_9^{\Theta^{Humidity}}(\{Normal\})$=0.3 $m_9^{\Theta^{Humidity}}(\Theta^{Humidity})$=0.7	$m_9^{\Theta^{Windy}}(\{True\})$=1	?

- *Let $\Theta^A = \Theta^{Outlook} \times \Theta^{Temperature} \times \Theta^{Humidity} \times \Theta^{Windy}$ be the global frame of discernment.*

- *The extension of the different bbas to the frame of discernment Θ^A is given in what follows:*

  - $m^{\Theta^{Outlook} \uparrow \Theta^A}(\{Sunny\} \times \Theta_{Temperature} \times \Theta_{Humidity} \times \Theta_{Windy})$*=0.6*
  - $m^{\Theta^{Outlook} \uparrow \Theta^A}(\{Overcast\} \times \Theta_{Temperature} \times \Theta_{Humidity} \times \Theta_{Windy})$*=0.4*
  - $m^{\Theta^{Temperature} \uparrow \Theta^A}(\Theta_{Outlook} \times \{Hot\} \times \Theta_{Humidity} \times \Theta_{Windy})$*=0.5*
  - $m^{\Theta^{Temperature} \uparrow \Theta^A}(\Theta_{Outlook} \times \{Mild\} \times \Theta_{Humidity} \times \Theta_{Windy})$*=0.5*
  - $m^{\Theta^{Humidity} \uparrow \Theta^A}(\Theta_{Outlook} \times \Theta_{Temperature} \times \{Normal\} \times \Theta_{Windy})$*=0.3*
  - $m^{\Theta^{Humidity} \uparrow \Theta^A}(\Theta^A)$*=0.7*
  - $m^{\Theta^{Windy} \uparrow \Theta^A}(\Theta_{Outlook} \times \Theta_{Temperature} \times \Theta_{Humidity} \times \{True\}$*=1*

- *The combination result of the extended bba through the conjunctive operator is:*
  - $m^{\Theta^A} = m^{\Theta^{Outlook}\uparrow\Theta^A} \mathbin{\bigcirc\!\!\!\!\!\cap} m^{\Theta^{Temperature}\uparrow\Theta^A} \mathbin{\bigcirc\!\!\!\!\!\cap} m^{\Theta^{Humidity}\uparrow\Theta^A} \mathbin{\bigcirc\!\!\!\!\!\cap} m^{\Theta^{Windy}\uparrow\Theta^A}$ *such that:*
    - $m^{\Theta^A}(Sunny, Hot, Normal, True) = 0.09$
    - $m^{\Theta^A}(Sunny, Hot, \Theta_{Humidity}, True) = 0.21$
    - $m^{\Theta^A}(Sunny, Mild, Normal, True) = 0.09$
    - $m^{\Theta^A}(Sunny, Mild, \Theta_{Humidity}, True) = 0.21$
    - $m^{\Theta^A}(Overcast, Hot, Normal, True) = 0.06$
    - $m^{\Theta^A}(Overcast, Hot, \Theta_{Humidity}, True) = 0.14$
    - $m^{\Theta^A}(Overcast, Mild, Normal, True) = 0.06$
    - $m^{\Theta^A}(Overcast, Mild, \Theta_{Humidity}, True) = 0.14$
- *For the classification process, we relied on decision trees given in Figure 3.1 and Figure 3.2 and we identify the corresponding leaves for each bbm:*

bbm	*Leaf (Figure 3.1)*	*Leaf (Figure 3.2)*
$m^{\Theta^A}(Sunny, Hot, Normal, True)$	L_6	L_1
$m^{\Theta^A}(Sunny, Hot, \Theta_{Humidity}, True)$	L_1, L_6	L_1
$m^{\Theta^A}(Sunny, Mild, Normal, True)$	L_7	L_1
$m^{\Theta^A}(Sunny, Mild, \Theta_{Humidity}, True)$	L_4, L_7	L_1
$m^{\Theta^A}(Overcast, Hot, Normal, True)$	L_6	L_2
$m^{\Theta^A}(Overcast, Hot, \Theta_{Humidity}, True)$	L_2, L_6	L_2
$m^{\Theta^A}(Overcast, Mild, Normal, True)$	L_7	L_2
$m^{\Theta^A}(Overcast, Mild, \Theta_{Humidity}, True)$	L_4, L_7	L_2

The, we compute the probability distribution over each class as follows:

- *Decision Tree corresponds to Figure 3.1:*
  - *Pr(Yes)=0.09 × 0.45 + 0.21 × 0.70 + 0.09 × 0.37 + 0.21 × 0.6 + 0.06 × 0.45+ 0.14 × 0.23 +0.06 × 0.37 + 0.14 × 0.6= 0.51*
  - *Pr(No)=0.09 × 0.55 + 0.21 × 0.30 + 0.09 × 0.63 + 0.21 × 0.40 + 0.06 × 0.55+ 0.14 × 0.77 +0.06 × 0.63 + 0.14 × 0.40= 0.49*

Accordingly, we can deduce that the most probable hypothesis is that customers will play Golf in the day D_9.

- *Decision Tree corresponds to Figure 3.2:*
  - $Pr(Yes)$= *0.09* × *0.95* + *0.21* × *0.95* + *0.09* × *0.95* + *0.21* × *0.95* + *0.06* × *0.86*+ *0.14* × *0.86* +*0.06* × *0.86* + *0.14* × *0.86*= *0.91*
  - $Pr(No)$= *0.09* × *0.048* + *0.21* × *0.048* + *0.09* × *0.048* + *0.21* × *0.048* + *0.06* × *0.14*+ *0.14* × *0.14* +*0.06* × *0.14* + *0.14* × *0.14*= *0.09*

From this results, we can induce that customers will play Golf in the day D_9.

3.3 Enhanced Evidential *k*-Nearest Neighbors (EE*k*-NN)

The *k* nearest neighbor classifier, firstly proposed by Fix and Hodges (1951), is also considered as one of the well commonly used classification techniques in the fields of machine learning and pattern recognition. The original *k*-NN version consists of assigning a query pattern to the majority class of its *k* nearest neighbors. The major shortcoming of this technique arises from learning a *k*-NN classifier with skewed class distributions, meaning that training instances with the most prevalent class may dominate the prediction of new query patterns due a large value of *k*. From this, numerous researchers have proven that the uncertainty about the class label of a given test pattern can be modeled through various uncertainty theories, particulary the belief function theory (Shafer, 1976).

Denœux (1995) has proposed an evidence theoretic *k*-NN (E*k*-NN) method relied on the belief function theory where each neighbor of a query pattern is regarded as a piece of evidence supporting some hypothesis concerning its class membership. The basic belief assignments obtained by all the *k* nearest neighbors have to be merged through the Dempster rule to get the final decision. An extended version of the E*k*-NN, called Evidential Editing *k*-NN (EE*k*-NN), has been introduced in (Jiao, Denœux, & Pan, 2015), where the label class of each training instance has to be represented by an evidential label to handle the uncertainty pervading the class labels. Despite their seriousness, neither the E*k*-NN nor the EE*k*-NN are able to handle data with evidential attributes. Inspired from these approaches, we suggest to develop an Enhanced Evidential *k*-Nearest Neighbors classifier for handling evidential data.

Let X be a training set described by N objects x_j (i.e. $j \in \{1,\ldots,N\}$) where each of them is described by n evidential attribute values $A = \{A_1,\ldots,A_n\}$ and a class label θ_j expressing with certainty its membership to one class in Θ.

Suppose that z is a new pattern to be classified on the basis of the information contained in the training set T. The idea consists of computing the distance between the test pattern z and each object x_j in X using a distance metric $d_{z,j}$. This distance has to be calculated as the sum of the absolute differences between the attribute values. More specifically, we have resorted to the Jousselme distance metric (A. Jousselme et al., 2001). A small value of $d_{z,j}$ reflects the situation that both instances z and x_j have the same class label and a large value of $d_{z,j}$ may reflect the situation of almost complete ignorance. The information concerning the label class of the instance z can be modeled through the belief function theory. So that, each training instance x_j provides an item of evidence $m^{(j)}(.|x_j)$ over Θ as follows:

$$\begin{aligned} &m^{(j)}(\{\theta_q\}|x_j) = \alpha\Phi_q(d_{z,j}) \\ &m^{(j)}(\Theta|x_j) = 1 - \alpha\Phi_q(d_{z,j}) \\ &m^{(j)}(A|x_j) = 0, \forall A \in 2^{\Theta}\backslash\{\Theta,\theta_q\} \end{aligned} \tag{3.25}$$

where $d_{z,j}$ is computed such as in Equation 2.20, θ_q is the class label of the instance x_j and α is a parameter such that $0 < \alpha < 1$. It has been proven by Denœux that a value of α equals 0.95 can yield good results. The decreasing function Φ_q, verifying $\Phi_q(0)$=1 and $lim_{d\rightarrow\infty}\Phi_q(d) = 0$, should be set to:

$$\Phi_q(d) = exp(-\gamma_q d^2) \tag{3.26}$$

where γ_q is a positive parameter relative to the class θ_q. It can be optimized using either an exact method or a linearization method (Zouhal & Denœux, 1998). The optimization process consists of minimizing the mean squared classification error over the whole training set X.

The final bba m^z can be obtained by merging the N bbas issued from the different training instances using the Dempster operator:

$$m^z = m^{(1)}(.|x_1) \oplus m^{(2)}(.|x_2) \oplus \ldots \oplus m^{(N)}(.|x_N) \tag{3.27}$$

As some training instances may be too far from z, only the k nearest neighbors should be considered to determinate the class membership. The final bba is obtained as follows as follows:

$$m^z = m^{(1)}(.|x_1) \oplus m^{(2)}(.|x_2) \oplus \ldots \oplus m^{(k)}(.|x_k) \tag{3.28}$$

Table 3.11: Distance between query instances

	Outlook	Temperature	Humidity	Windy	Total Distance	Average Distance
$d(D_1,D_9)$	0.4	0.25	0.35	0	1	0.25
$d(D_2,D_9)$	0.2	0.15	0.28	1	1.63	0.40
$d(D_3,D_9)$	0.6	0.5	0.14	0.90	2.14	0.53
$d(D_4,D_9)$	0.75	0.5	0.46	1	2.71	0.67
$d(D_5,D_9)$	0.87	0.68	0.49	0.89	2.94	0.73
$d(D_6,D_9)$	0.52	0.86	0.53	0	1.92	0.48
$d(D_7,D_9)$	0.40	0.04	0.28	0.95	1.67	0.41
$d(D_8,D_9)$	0.4	0.7810	0.63	1	2.81	0.70

The test pattern is then assigned to the class with the maximum pignistic probabilit of m^z:

$$\theta_{j^*}(z) = argmax_{\theta_q \in \{\theta_1,\dots,\theta_c\}} BetP(\{\theta_q\}) \tag{3.29}$$

where $BetP(\{\theta_q\})$ corresponds to the pignistic probablity of the hypothesis θ_q associated to the bba m^z.

Example 3.7. *Let us continue with the evidential training instances given in Table 3.2 and the test instance presented in Table 3.10. We try to predict the label class of the query instance through our proposed EEk-NN, when following these steps:*

- *Assume that* $\gamma_{y}es$ *and* $\gamma_{N}o$ *are equal respectively to 0.47 ad 0.53. We calculate the distance between the query instance* D_9 *and each training instance. The results are presented in Table 3.11.*
- *Suppose that k equals 3, then the three nearest neighbors of* D_9 *are* D_1*,* D_2 *and* D_7*. Accordingly, the output labels of these three instances will be modeled through belief functions as follows:*

D_1	$m_1(\{Yes\}) = 0.95 \times e^{-0.47*0.25^2}$=*0.90* $m_1(\Theta) = 1 - m_1(\{Yes\})$=*0.10*
D_2	$m_2(\{Yes\}) = 0.95 \times e^{-0.47*0.41^2}$= *0.87* $m_2(\Theta) = 1 - m_2(\{Yes\})$= *0.13*
D_7	$m_7(\{No\}) = 0.95 \times e^{-0.53*0.42^2}$= *0.84* $m_7(\Theta) = 1 - m_7(\{No\})$= *0.16*

- *Besides, we have to merge these bbas through the Dempster operator as follows:*

  $$m_9 = m_1 \oplus m_2 \oplus m_7$$

- *Thus:*

  $$m_9(\{Yes\}) = 0.92$$
  $$m_9(\{No\}) = 0.07$$
  $$m_9(\Theta) = 0.01$$

- *From this, the pignistic probability relative to m_9 is equal to:*

  $$BetP(Yes) = 0.93$$
  $$BetP(No) = 0.07$$

  According to the obtained results, it is more probable that customers play golf in D_9.

3.4 Comparative study

This Section is devoted to examining the performance of our three proposed evidential machine learning classifiers. In what follows, we details our experimentation settings and results.

3.4.1 Experimentation settings

To evaluate the performance of our three evidential classifiers, we have relied on some numerical and mixed real world databases acquired from the UCI machine learning databases (Murphy & Aha, 1996), where some of them are characterized by the presence of missing values. Table 3.12 provides a description of the used databases. In a practical point of view, missing values have to be imputed and continuous variables have usually to be discretized into bins. However, the uncertainty introduced by missing values imputation and continuous variables discretization have to be addressed. Herein, we propose to generate evidential databases from the mentioned ones. That is, the missing values will be represented by vacuous bbas and symoblic attributes have to be expressed through certain bbas. With regards to continuous variables, they have been transformed into beliefs using the Evidential c-Means approach (ECM) (Masson & Denœux, 2008; Samet, Lefèvre, & Ben Yahia, 2016).

Table 3.12: Description of databases

Databases	Heart	Japanese	Vote records	Hepatitis	Wine	Thoracic Surgery
Total instances	270	690	435	155	178	470
Total attributes	13	15	16	19	13	17
Missing values	No	No	Yes	Yes	No	No
Number of classes	3	2	2	2	3	2

For the evaluation process, we relied on some standard information retrieval measures: the Percentage of Correctly Classification (PCC), the recall and the precision. Let Figure 2.3 illustrates the confusion matrix of a c class classification problem.

The PCC is calculated as the percentage of well classified instances. It corresponds to the recogration rate computed following to Equation 2.40.

The recall measure, also refereed to as TP rate or sensitivity, is the proportion of correctly classified positive instances, with respect to all positive instances. The sensitivity of each class $\theta_i \in \Theta$ is computed as follows:

$$S_i = \frac{N_{i,i}}{\sum_{j=1}^{c} N_{i,j}} \tag{3.30}$$

and the sensitivity over all classes will be set to:

$$S = \frac{1}{c}\sum_{i=1}^{c} S_i \tag{3.31}$$

The precision is computed as the ratio of the number of correctly classified positive instances, with respect to the total number of predicted positive instances. The precision of each class $\theta_i \in \Theta$ is computed as follows:

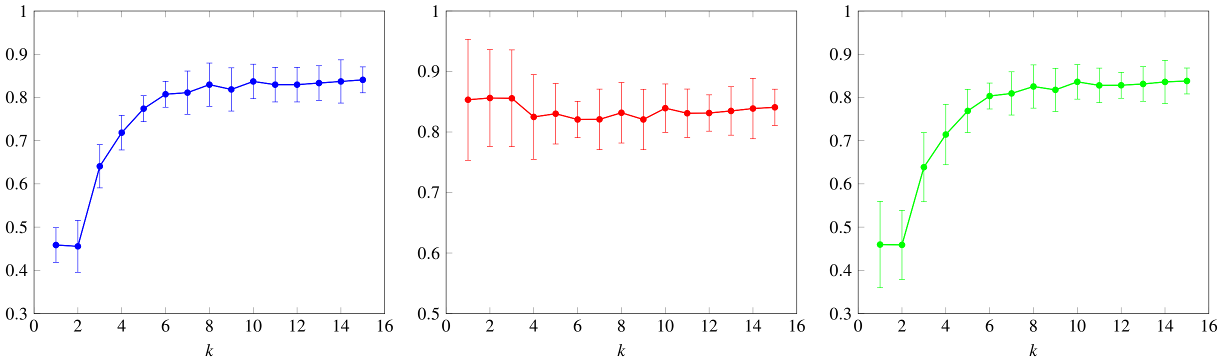
$$P_i = \frac{N_{i,i}}{\sum_{j=1}^{c} N_{j,i}} \tag{3.32}$$

and the precision over all classes will be set to:

$$P = \frac{1}{c}\sum_{i=1}^{c} P_i \tag{3.33}$$

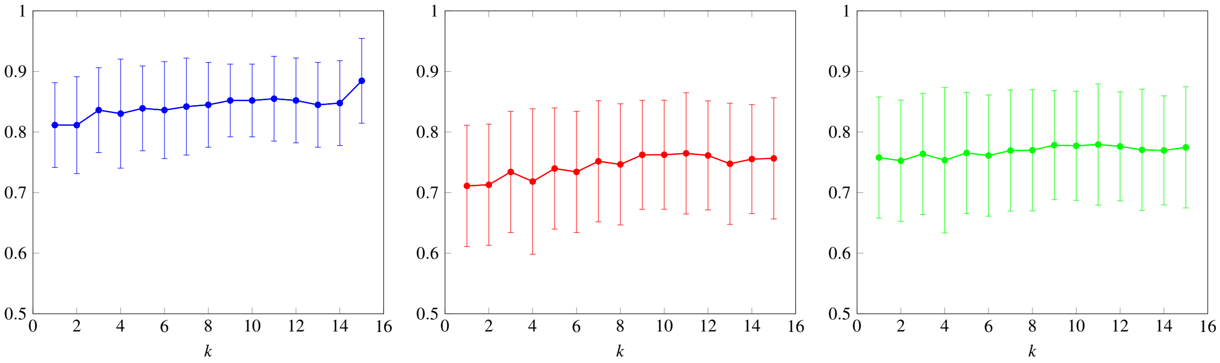
3.4.2 Results discussions

Following a 5-fold cross validation approach, we firstly present from Figure 5.1 to Figure 3.8, the PCC, the recall and the precision results relative to our EEk-NN classifier for all k in $\in \{1,\ldots,15\}$. knowing that in the literature we are restricted to impair values of k, herein, we aim to test several values of k in the purpose of studding the impact of the k values on the classification performance and the identifying the best vale of k in the range [1,15]. As remarked from these figures, the performance of our EEk-NN classifier varies according to the value of k. For the heart database the best PCC and precision values are yielded for k equals 15, while the recall value is achieved where k equals 2.



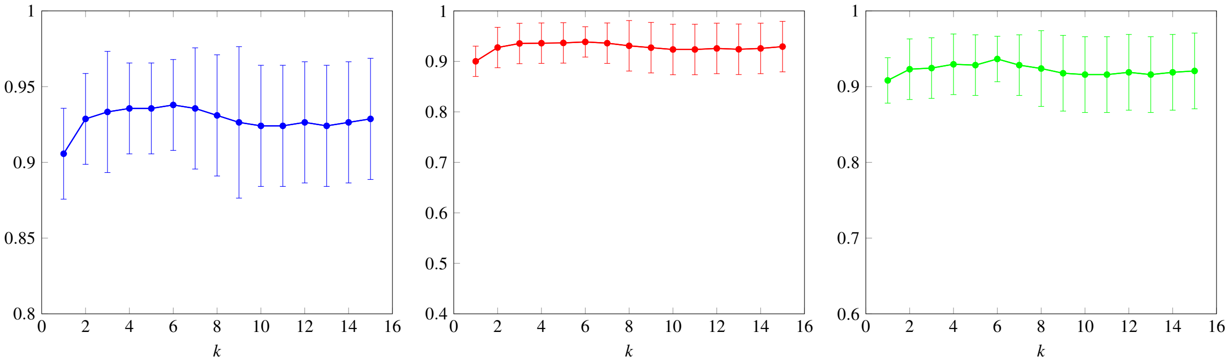
(a) PCC (b) Recall (c) Precision

Figure 3.3: Results for Heart database



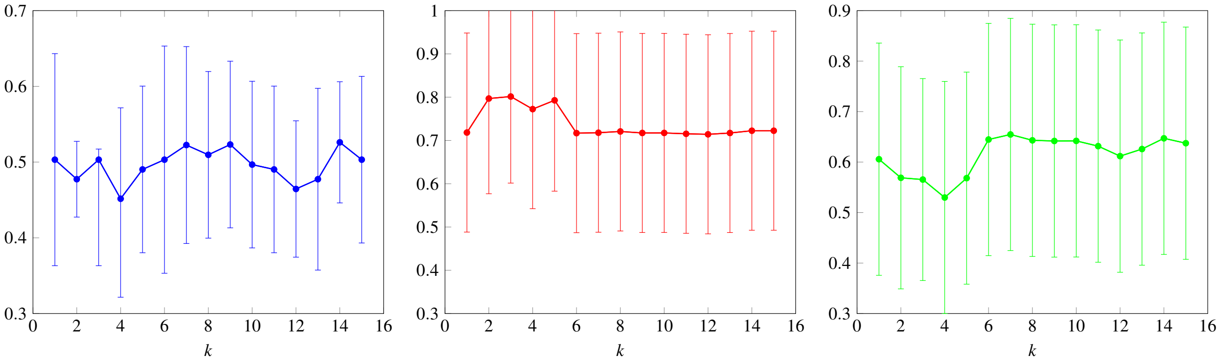
(a) PCC (b) Recall (c) Precision

Figure 3.4: Results for Japanese database



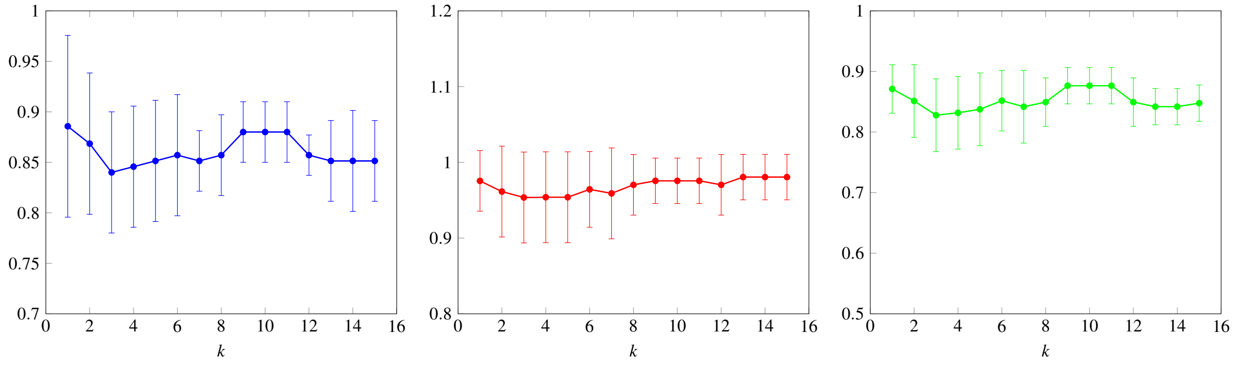
(a) PCC (b) Recall (c) Precision

Figure 3.5: Results for Vote Records database



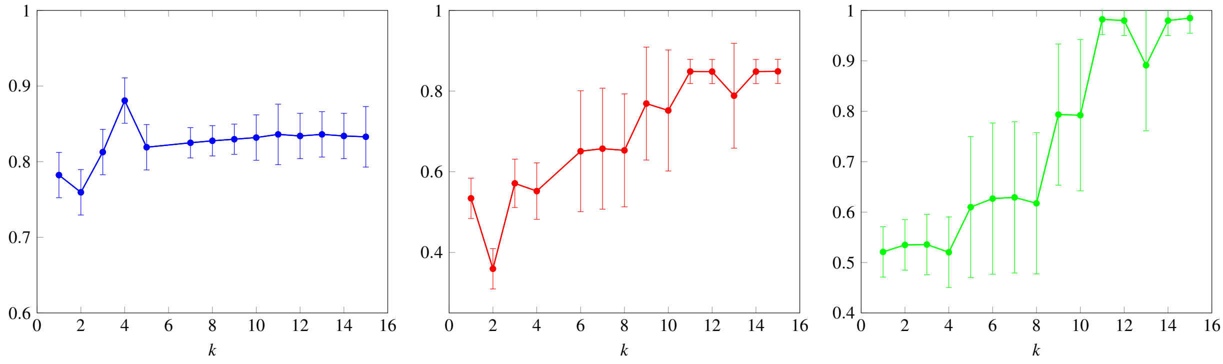
(a) PCC (b) Recall (c) Precision

Figure 3.6: Results for Hepatitis database



(a) PCC (b) Recall (c) Precision

Figure 3.7: Results for Wine database



(a) PCC (b) Recall (c) Precision

Figure 3.8: Results for Thoracic Surgery database

Table 3.13: Comparative results in terms of the PCC criterion (%)

Bases	EE*k*-NN	Decision trees	
		Gain Ratio	Diff Ratio
Heart	**84.07** ± **0.03** *k*={15}	80.04 ± 0.15	78.32 ± 0.23
Japanese	**85.50** ± 0.07 *k*={11}	84.98 ± 0.01	82.63 ± 0.03
Vote Records	**93.79** ± **0.03** *k*={6}	89.86 ± 0.06	91.24 ± 0.12
Hepatitis	**52.25** ± **0.13** *k*={7}	49.08 ± 0.17	50.02 ± 0.26
Thoracic Surger	**83.61**± **0.03** *k*={11,13}	83.01 ± 0.2	81.17 ± 0.07
Wine	**88.57** ± **0.09** *k*={1}	79.82 ± 0.18	85.11 ± 0.26

Table 3.14: Comparative results in terms of the recall criterion (%)

Bases	EE*k*-NN	Decision trees	
		Gain Ratio	Diff Ratio
Heart	**85.62**± **0.08** **k={2}**	78.02± 0.17	75.64 ± 0.01
Japanese	76.48± 0.1 k={11}	**82.42** ± **0.26**	81.45 ± 0.03
Vote Records	92.93 ± 0.05 k={15}	85.56 ± 0.38	**98.56** ± **0.01**
Hepatitis	**80.15** ± **0.02** **k={3}**	47.42 ± 0.23	48.49 ± 0.03
Thoracic Surger	**84.59±0.03** **k={15}**	79.78 ± 0.07	79.17 ± 0.1
Wine	**98.05**± **0.03** **k={13,14,15}**	75.46 ± 0.13	82.96 ± 0.22

The performance of our proposed *EEk*-NN classifier with the best value of *k* will be compared to our two proposed decision tree classifiers. The comparative results are given from Table 3.13 to Table 3.15.

Table 3.15: Comparative results in terms of the precision criterion (%)

Bases	EE*k*-NN	Decision trees	
		Gain Ratio	Diff Ratio
Heart	**83.81 ±0.03** **k={15}**	79.66 ± 0.03	77.76 ± 0.1
Japanese	77.94±0.1 k={11}	**83.12 ± 0.24**	81.89 ± 0.04
Vote Records	**93.65 ± 0.03** **k={6}**	87.89 ±0.02	90.07 ± 0.23
Hepatitis	**65.47 ± 0.23** **k={7}**	48.26 ± 0.17	49.05 ± 0.04
Thoracic Surger	**98.18 ±0.03** **k={15}**	82.44 ± 0.03	81.05 ± 0.19
Wine	**87.65±0.03** **k={9,10,11}**	78.56 ± 0.04	84.25 ± 0.01

The results presented in Table 3.13, Table 3.14 and Table 3.15 prove the performance of our EE*k*-NN classifier comparatively with our decision tree approaches in terms of the PCC, the recall and the precision criteria for almost all databases and almost all values of *k*. This is justified by the fact that our proposed decision tree approaches generate trees with a huge number of branches. The occurred over-fitting has a greatest influence in the classification performance. Another point to be highlighted is that the EE*k*-NN classifier performs faster than the decision tree versions.

3.5 Conclusion

In this Chapter, we have proposed three machine learning classifiers for handling real world data with evidential attributes. The two first ones are extensions of the classical decision tree classifier, while the third one, referred to as Enhanced Evidential *k*-NN, is an extension of the standard *k*-NN classifier. To pick out the most efficient algorithm among these proposed ones, we have carried out a comparative study when relied on the PCC, the recall and the precision as assessment criteria. The obtained results have proven the efficiency of the EE*k*-NN comparatively with the decision tree versions across the different databases. To this end, in the next Chapter, we develop an ensemble EE*k*-NN classifier through feature subspaces. More precisely, we propose a rough set based ensemble EE*k*-NN classifiers.

Chapter 4

A selective ensemble EEk-NN classifiers through rough set reducts

Contents

4.1 Introduction **75**

4.2 Rough sets: Fundamental concepts **76**

4.3 A rough set based selective ensemble EEk-NNs **80**

4.3.1 Reduct extraction from data with evidential attributes . 80

4.3.2 Reduct selection for ensemble EEk-NN classifiers . . . 82

4.4 Experimentation settings and results **88**

4.4.1 Ensemble size 88

4.4.2 Reduct diversity 90

4.5 Conclusion **91**

4.1 Introduction

The construction of a good ensemble classifier has become a vital need thanks to its ability for yielding appropriate decisions. One solution consists of constructing an ensemble of classifiers through feature subspaces (Breiman, 2001; Lienemann et al., 2007; Skurichina & Duin, 2002). The main key underlying this approach concerns the generation of the most suitable feature subsets, meaning the minimal attribute subsets that are diverse as much as possible and allowing the same classification ability as the original attribute set. In this context, classifier ensembles through rough set reducts have been very well studied for quite some times (Shi, Ma, et al., 2011; Shi, Xi, et al., 2011; Wang et al., 2010; Saha et al., 2008).

Since rough set ensemble classifiers for addressing imperfect knowledge, notably evidential ones, have not attracted the great attention till now, we propose, in this thesis, to construct a rough set based ensemble for handling such a kind of data. In analogy with the standard case, the construction of a good ensemble requires two main steps. the selection of the best individual classifiers (i.e. meaning the most suitable reducts) and the combination of these classifiers. In this chapter, we only focus on the classifier selection. Accordingly, we propose a novel framework for constructing individual base classifiers trained with the most suitable reducts and enabling the generation of a successfully ensemble of the EE*k*-NN classifiers. So, we start by highlighting the basic concepts behind the rough set theory and we present then our novel framework for generating and selecting appropriate reducts enabling the construction of a good ensemble EE*k*-NN classifiers.

4.2 Rough sets: Fundamental concepts

The rough set theory, proposed by Pawlak (1998), is one efficient way for dealing with various application problems, including feature subspaces. It allows to generate the smallest subsets of relevant features, also called reducts, enabling the same discrimination as the original attribute set. In practical terms, a data set has to be represented through a Decision Table (DT) which is defined as a pair $DT = (X, A \cup \{y\})$. The universe $X = \{x_1, \ldots, x_N\}$ reflects a non-empty finite set of N objects, $A = \{A_1, \ldots, A_n\}$ is a non-empty finite set of n condition attributes with values $V(x_j) = \{V_1(x_j), \ldots, V_n(x_j)\}$ for each object x_j and $y \in \Theta = \{\theta_1, \ldots, \theta_c\}$ corresponds to the decision attribute value (Yao & Zhao, 2009). Let B denotes a subset of attributes (i.e. $\mathrm{B} \subseteq A$), an indiscernibility relation, denoted $IND(B)$, is defined by the following $\forall\, k = \{1, \ldots, n\}$:

$$IND(B) = \{(x_i, x_j) \in X \times X | \forall A_k \in B, V_k(x_i) = V_k(x_j)\} \tag{4.1}$$

Example 4.1. *Let us continue with Example 3.1 to illustrate how to define an indiscernibility relation from a decision table. we consider the following three non-*

empty subsets of the conditional attributes: $\{Outlook\}$, $\{Outook, Temperature\}$ *and* $\{Outlook, Temperature, Humidity, Windy\}$.

- $IND(Outlook) = \{\{D_1, D_2, D_8\}, \{D_3, D_7\}, \{D_4, D_5, D_6\}\}$.
- $IND(Outlook, Temperature) = \{\{D_1, D_2\}, \{D_3\}, \{D_4\}, \{D_5, D_6\}, \{D_7\}, \{D_8\}\}$.
- $IND(Outlook, Temperature, Humidity, Windy) =$ $\{\{D_1\}, \{D_2\}, \{D_3\}, \{D_4\}, \{D_5\}, \{D_6\}, \{D_7\}, \{D_8\}\}$.

Assuming that U is a subset of the universe X, the B-lower of U denoted by $\underline{B}(U)$ and the B-upper of U denoted by $\overline{B}(U)$ are illustrated in Figure 4.1 and are computed as follows:

$$\underline{B}(U) = \{x_j | [x_j]_B \subseteq U, x_j \in U\} \tag{4.2}$$

and

$$\overline{B}(U) = \{x_j | [x_j]_B \cap U \neq \emptyset, x_j \in U\} \tag{4.3}$$

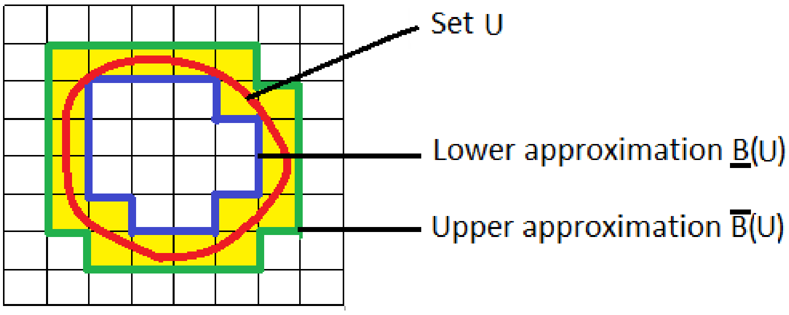


Figure 4.1: The illustration of the set approximation

Based on the knowledge in B, objects in $\underline{B}(U)$ can be with certainty classified as members of U and objects in $\overline{B}(U)$ can be only classified as possible members of U.

Let B and y be equivalence relations over U, then the positive, negative and boundary regions can be defined:

$$POS_B(y) = \bigcup_{U \in X/y} \underline{B}(U) \tag{4.4}$$

$$NEG_B(y) = U - \bigcup_{U \in X/y} \overline{B}(U) \tag{4.5}$$

$$BND_B(y) = \bigcup_{U \in X/y} \overline{B}(U) - \bigcup_{U \in X/y} \underline{B}(U) \tag{4.6}$$

The positive region contains all objects of U that can be classified to classes of U/y using the information in attributes B. The boundary region, $BND_B(y)$, is the set of objects that can possibly, but not certainly, be classified in this way. The negative region, $NEG_B(y)$, is the set of objects that cannot be classified to classes of U/y.

Example 4.2. *Let us continue with the previous example Example 2.3. An illustrative example of the above mentioned calculations is given in what follows where B=$\{$Outlook, Temperature$\}$:*

- $POS_B(y) = \bigcup\{\{D_1,D_2,D_5,D_6\},\{D_3,D_4,D_7\}\}$
  $=\{D_1,D_2,D_3,D_4,D_5,D_6,D_7\}$
- $NEG_B(y)$= U-$\bigcup\{\{D_1,D_2,D_5,D_6\},\{D_3,D_4,D_7\}\} = \{D_8\}$
- $BND_B(y)$= $\{\emptyset\}$

Retain the attributes that preserve the positive region is regarded as an effective alternative for feature reduction. Since this approach may yield several feature subsets, only minimal ones, refers to as reducts, have to be chosen for solving a given machine learning problem. A subset B is a reduct of A with respect to the decision attribute y, if B is minimal and:

$$Pos_B(y) = Pos_A(y) \tag{4.7}$$

Another efficient solution for reduct extraction is the discernibility function (Skowron & Rauszer, 1992). It consists firstly of computing a discernibility matrix DM from a given decision table DT. The entries of the discernibility matrix DM are computed as follows:

$$DM(x_i,x_j) = \{A_k \in A | V_k(x_i) \neq V_k(x_j) \text{ and } y_i \neq y_j\} \ \forall \ i,j = \{1,\dots,N\} \tag{4.8}$$

Once the discernibility matrix *DM* is computed, we can define the discernibility function as follows:

$$f(DM) = \wedge\{\vee(DM(x_i,x_j))|\forall x_i,x_j \in X, DM(x_i,x_j) \neq \emptyset\} \tag{4.9}$$

The discernibility function has to be converted from a conjunctive normal form into a disjunctive normal form for picking out all possible reducts.

Example 4.3. *Let us continue with Example 3.1 and try to compute the discrnibility matrix. Assume that O, T, H and W reflect respectively the attribute Outlook, Temperature, Humidity and Windy. The corresponding discernibility matrix is given in Table 4.1 The discrnibility function is then computed as follows:*

Table 4.1: An example of a discernibility matrix

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
D_1	-	-	-	-	-	-	-	-
D_2	O,T,H	-	-	-	-	-	-	-
D_3	T,H,W	T,H	-	-	-	-	-	-
D_4	H,W	H	H,W	-	-	-	-	-
D_5	W	-	-	O	-	-	-	-
D_6	W	W	W	O,W	O,T,H	-	-	-
D_7	-	W	O	-	T,H,W	T,H	-	-
D_8	O,H,W	O,H	H,W	T,H,W	-	W	-	-

$$\begin{aligned} f(DM) &= \wedge\{\vee(\{O\},\{H\},\{W\},\{O,W\},\{O,H\},\{H,W\},\{T,H\},\{T,H,W\},\{O,H,W\},\{O,T,H\})\} \\ &= \{O \wedge H \wedge W\} \end{aligned}$$

From this example, we can deduce that there is a unique reduct for the data given in Table 3.1 (i.e. R= {O, H, W}).

In spite of their great success, this approach is very costly and it is impractical for medium sized or large sized data sets. In light of this shortcoming, several heuristics have been introduced (Johnson, 1973; Komorowski, Øhrn, & Skowron, 2002)

4.3 A rough set based selective ensemble EE*k*-NNs

Almost all existing datasets are described by redundant and irrelevant features that could adversely affect the classification performance, the computation time and the memory resources. The rough set theory is widely used for relevant feature extraction and it has been regarded as a novel way for producing successful ensemble classifiers, we aim throughout this dissertation to construct a rough set based ensemble EE*k*-NN classifiers for addressing data with evidential attributes. We define the notion of reduct with evidential attributes and we develop a new rough set reduct method for picking out all possible reducts within an evidential environment. In a traditional way, ensemble classifiers have to be constructed with all generated reducts. However, we could very well wind up with hundreds or even thousands of reducts. Thus, we propose to select the most suitable reducts for an ensemble of EE*k*-NN classifiers. Figure 4.2 provides a description of our novel rough set reducts based selective ensemble.

4.3.1 Reduct extraction from data with evidential attributes

Reduct computation has been proven as an NP-hard problem which has led to the introduction of several heuristics. The Rosetta software is well-known to be among the most effective methods for reducts generation. It includes a set of algorithms for extracting multiple reducts. An example includes the SAVGenetic Reducer (El-Monsef et al., 2003), a genetic algorithm for picking out approximate reducts. As our aim purpose is to address data with evidential attributes, we propose to extend the SAVGenetic algorithm to a distance-based definition to handle evidential attributes when maintaining the used fitness function. In analogy with the standard version, our belief SAVGenetic reducer starts by calculating a discernibility matrix Λ' from the evidential data (Trabelsi, Elouedi, & Lefèvre, 2016a). Assume that X=$\{x_1,\ldots,x_N\}$ is a given data set with N objects. Each object x_i $(i \in \{1,\ldots,N\})$ is described by n evidential attributes $A = \{A_1,\ldots,A_n\}$ and a certain class label $y_i \in \Theta$=$\{\theta_1,\ldots,\theta_c\}$. Note that each attribute A_k (i.e. $k \in \{1,\ldots,n\}$) has a domain of discrete values denoted by Θ^{A_k}, the entries of the discernibility matrix Λ' are set to:

$$\Lambda'(x_i,x_j) = \{A_k \in A | dist(m_i^{\Theta^{A_k}}, m_j^{\Theta^{A_k}}) > T \text{ and } y_i \neq y_j\} \tag{4.10}$$

where $m_i^{\Theta^{A_k}}$ states the bba assigned to the attribute A_k of the object x_i, T refers to

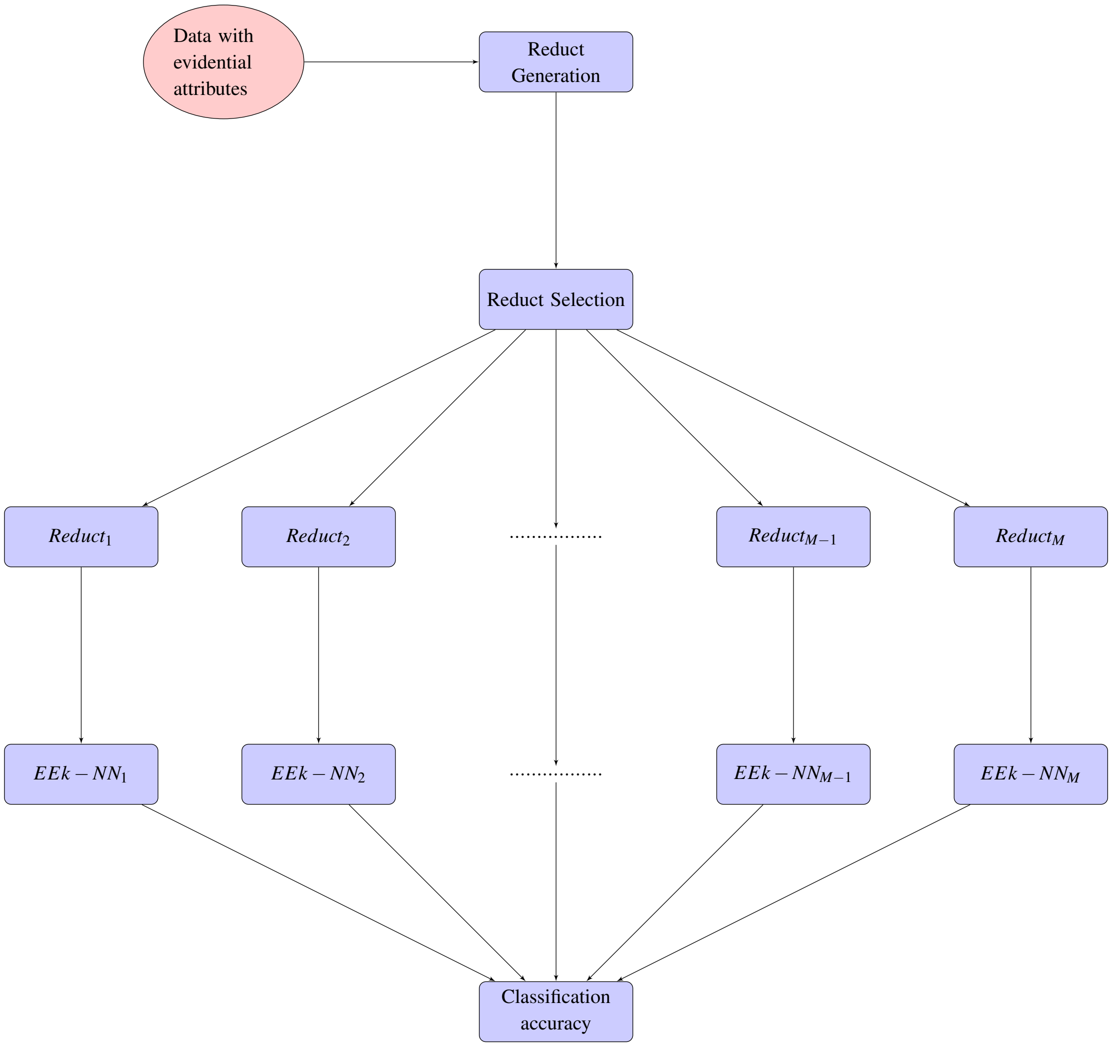


Figure 4.2: A rough set based ensemble for data with evidential attributes

a tolerance threshold (i.e. T is set to 0.4 with the aim of maximizing the search space) and *dist* reflects the Jousselme distance (A. Jousselme et al., 2001).

The process of extracting reducts through a discernibility matrix is regarded as a set cover problem. It consists of finding the minimal hitting sets [1] form the non empty sets of the obtained discrenability matrix. Since the minimal hitting set is an NP-hard problem, we relied on the genetic algorithm for picking out approximate hitting sets, meaning approximate reducts. Suppose that ζ' contains the non empty sets of Λ', the fitness function corresponds to our genetic algorithm for each subset $B \in 2^n$ is set to:

$$f(B) = (1-\alpha) \times \frac{|A| - |B|}{|B|} + \alpha \times \varepsilon, \frac{|[F \in \zeta' | F \cap B \neq \emptyset]|}{|\zeta'|} \tag{4.11}$$

The fitness function $f(B)$ rewards not only subsets that are hitting sets (i.e. meaning subsets having a non empty intersection with all elements of the discernability matrix) but also subsets with shortest size. Herein, $\alpha \in [0,1]$ refers to the adaptive weighting between the two parts. In our case, we have set α to 0.5.

4.3.2 Reduct selection for ensemble EE*k*-NN classifiers

As already mentioned, we could end up with several reducts. The process of constructing ensemble systems with all generated classifiers is extremely costly especially for high dimensional databases. An alternative solution consists of selecting the most appropriate reducts for ensemble learning. We present, throughout this Section, three approaches enabling to pick out the most suitable reducts for an ensemble of EE*k*-NN classifiers.

Select Diverse Reducts (DR)

One of the main keys for constructing a successful rough set ensemble is to ensure a good diversity between the chosen reducts. Getting inspiration from (Debie et al., 2013), we propose a new heuristic for selecting diverse reducts from the

---

[1] A hitting set of a given multiset ζ' of elements from 2^n is a set $B \subseteq n$ such that the intersection between B and every set in ζ' is non-empty. The set $B \in HS(\zeta')$ (i.e HS denotes the hitting sets) is a minimal hitting set of ζ' if any of its elements are removed

pool of generated ones (see Algorithm 4.1) when starting with the smallest reduct instead of applying a random choose. Once the first reduct R_1 is picked out, our algorithms computes the diversity between the chosen reduct R_1 and the remaining ones using Algorithm 4.2. The reduct which has to be chosen is the one with the highest diversity degree. As in (Debie et al., 2013), the diversity measure will be computed as the inverse of the average similarity between a candidate reduct R_j and the M' chosen ones (i.e *RED_Chosen*):

$$Div_j = 1 - \frac{1}{M'} \sum_{R_i \in RED\_Chosen} \frac{|R_j \cap R_i|}{|R_j \cup R_i|} \tag{4.12}$$

The most diverse reduct R_j will then be chosen for constructing the ensemble system and it will be removed from the current reduct set *RED*. This process will be repeated until at most *M* reducts are selected or the reduct pool *RED* is empty. Regarding the time complexity, it depends mainly on the number of generated reducts (i.e $O(M \times |RED|)$).

Algorithm 4.1 Select diverse reducts (RD)

```
input: A pool of reducts $RED$, $M$ is the maximum number of chosen reducts
output: $M'$ diverse reducts
RED_Chosen ← $\emptyset$
$R_1 = \min_{R \in RED} cost(R)$
$RED\_Chosen \leftarrow \{RED\_Chosen, R_1\}$
$M'=1$
$RED = RED - R_1$;
While $M' <$ M or $isEmpty(RED) = false$ Do
    $Div \longleftarrow ReductDiversity(RED\_Chosen, RED)$ {%Computed through Algorithm 4.2}
    $R\_best = arg \max_{R_j \in RED} Div_j$
    $RED\_Chosen \leftarrow \{RED\_Chosen, R\_best\}$
    $RED = RED - R\_best$;
    $M' = M' + 1$
end while
```

Accuracy-Diversity Assessment Function for reduct selection (AD-AF)

The study conducted by Opitz and Maclin (1999) has demonstrated that both the accuracy and the diversity of base classifiers may improve the performance of an

Algorithm 4.2 ReductDiversity(*RED_Chosen*,*RED*)

```
input: A set of candidate reducts $RED$ and selected reducts $RED\_Chosen$
output: Diversity between reducts $Div$
for $j$= 1 to $|RED|$
    $Sim_j$=0;
    for each $R_i \in RED\_Chosen$
        $Sim_j$= $Sim_j$ + $\frac{|RED_j \cap R_i|}{|RED_j \cup R_i|}$
    end for
    $Div_j$ = 1- $\frac{Sim_j}{|RED\_Chosen|}$
end for
```

ensemble system. That is, a good ensemble of classifiers have to be constructed on the basis of accurate individual classifiers that are diverse as much as possible.

As already stated, a rough set based ensemble classifiers has been viewed for some years as a valid alternative for ensuring good diversity between the base classifiers. Herein, we propose to construct a performant classifier ensemble by assessing a good diversity between the attribute sets and making a trade-off between the diversity and the accuracy of each individual classifier. More precisely, we relied on the similarity-based diversity metric (i.e using Equation 4.12) for achieving the diversity between the the attribute sets and we relied on the assessment function, proposed by Opitz (Opitz & Maclin, 1999), to extract the most suitable reducts based on the predictions of the resulting classifier. Opitz's assessment function for a candidate classifier f_j constructed through a reduct R_j is ste to:

$$Fitness(f_j, Ens\_Cls) = Accuracy(f_j, Ens\_Cls) + \omega \times Diversity(f_j, Ens\_Cls) \tag{4.13}$$

where *Ens_Cls* states the current ensemble of classifiers, and ω corresponds to the parameter that balances *Accuracy* and *Diversity*. Concerning the parameter ω, it has to be adjusted automatically for maximizing the fitness function value. To put it more clearly, we keep the value of ω when *Fitness* is increasing, we increase it if *Accuracy* is stable and *Diversity* is decreasing and we decrease it if *Accuracy* is decreasing and *Diversity* is stable. It will be set to 1 as the initial value and the changing amount of ω will be set to 10% based on its current value. With regards to the diversity measure, it is important to note that there are several classifier diversity measures. Kuncheva and Whitaker (2003) have distinguished pairwise and non-pairwise diversity measures. The choice of the most convenient one remains unanswered question. Herein, we relied on the disagreement measure (see Equation 1.4), which is a pairwise one, for computing classifier diversity. Regarding the $Accuracy(f_j, Ens\_Cls)$, it reflects the average accuracy of the individual classifiers. Considering that $M'= |Chosen\_RED|$ is the number of actually selected reducts. The $Accuracy(f_j, Ens\_Cls)$ is computed as the average of the individual classifiers and it is obtained from the recognition rate (see Equation 2.40 as follows:

$$Accuracy(f_j, Ens\_Cls) = \frac{\sum_{i=1}^{M'} Recognition\_Rate_i + Recognition\_Rate_j}{M'+1} \tag{4.14}$$

Our method differs from Opitz's approach in the way that we take into consideration the accuracy between reducts in addition to the diversity and the accuracy of individual classifiers. Our proposed framework is detailed in Figure 4.3 and Algorithm 4.3.

We start by retrieving the reduct R_1 with the lowest cost [2] and constructing the first $EE\text{k}-NN_1$ classifier (i.e. $RED\_Chosen = \{R_1\}$ and $Ens\_Cls = \{EE\text{k}-NN_1\}$). It calculates then the diversity Div_j between the current selected reducts RED_Chosen and each reduct $R_j \in RED$ using Equation 4.12. Reducts R_j with a diversity measure smaller then a threshold S have to removed from the reduct pool RED. Each candidate reduct $R_j \in RED$ will be evaluated using Equation 4.13. The reduct R_k enabling the highest fitness function has to selected for constructing our ensemble learning (i.e. $Ens\_Cls = \{Ens\_Cls, EEk-NN_k\}$). This process has to be repeated until at most a number M of reducts is reached or the current reduct pool RED is empty.

The key point behind the threshold S is to guarantee the selection of diverse reducts. So that, if S equals 0 then all the generated reducts have to be considered as candidates and if S equals 1 there is there is no reduct candidate. In our experimentation parts, we have set S to 0.7 for reducing the search space). However, a further studies have to be done for optimizing the value of S that affect the number of candidate reducts to be assessed through the fitness function.

---

[2] The smallest reduct

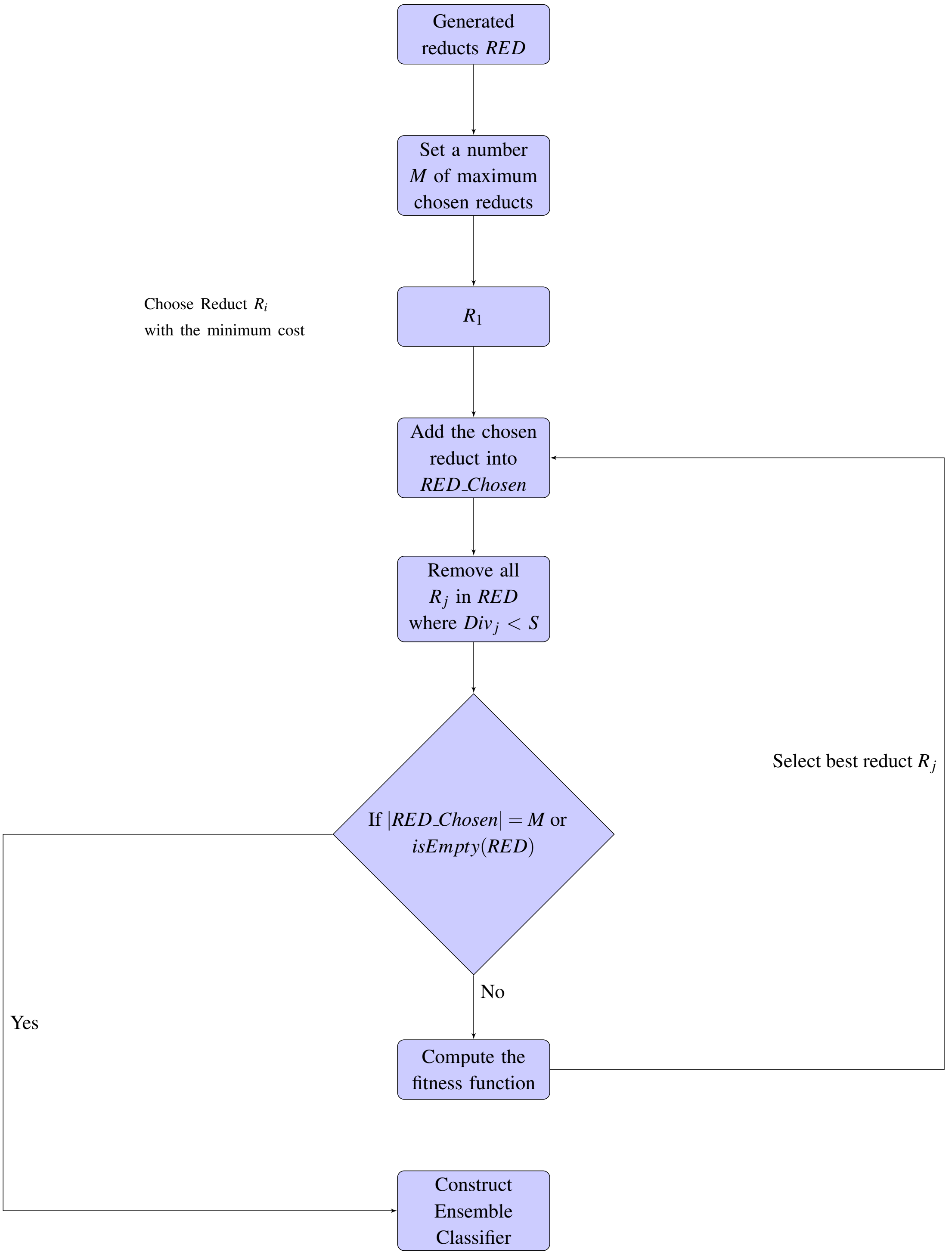

Generated reducts RED
Set a number M of maximum chosen reducts
Choose Reduct R_i with the minimum cost
R_1
Add the chosen reduct into RED_Chosen
Remove all R_j in RED where $Div_j < S$
If $|RED\_Chosen| = M$ or $isEmpty(RED)$
Select best reduct R_j
No
Yes
Compute the fitness function
Construct Ensemble Classifier

Figure 4.3: Reduct selection for ensemble learning

Algorithm 4.3 A reduct selection approach based on the Accuracy-Diversity assessment function

```
input: A pool of reducts $RED$, $m$ is the maximum Number of chosen reducts
output: Chosen $m'$ diverse reducts
$RED\_Chosen \leftarrow \emptyset$
$Ens\_Cls \leftarrow \emptyset$
$R_1$=$\min_{R \in RED} cost(R)$
$RED\_Chosen \leftarrow \{RED\_Chosen, R_1\}$
$M'$=1
$RED$= $RED$ - $R_1$;
$Ens\_Cls \leftarrow \{Ens\_Cls, f_1\}$
Repeat
    $Div \longleftarrow ReductDiversity(RED\_Chosen, RED)$ {% Computed through Algorithm 4.2}
    $RED\_New \longleftarrow R_j \in RED$ with $Div_j > S$
    $RED = RED\_New$
    Choose a new reduct $R_j$ from $RED$ satisfying:
    $Fitness(f_j, Ens\_Cls) = \max_{R_k \in RED}(Fitness(f_k, Ens\_Cls))$
    $Ens\_Cls \leftarrow \{Ens\_Cls, f_j\}$
    $RED\_Chosen \leftarrow \{RED\_Chosen, R_j\}$
    $RED$=$RED$-$R_j$
    $M'$=$M'$+1
until $M' = M$ or $isEmpty(RED) = true$
```

Ensemble Accuracy Assessment Function for reduct selection (EA-AF)

The wrapper approach, using the classifier accuracy as feature selection criterion, has been successfully used for solving several pattern recognition problems. In fact, it allows to pick out the feature subset that achieves the greatest classification accuracy. Herein, we follow the same process as the previous presented approach but we relied on the ensemble accuracy as a fitness function for extracting the most appropriate reducts for an ensemble of EEk-NN classifiers. The fitness function is set to:

$$Fitness(f_j, Ens\_Cls) = EnsAcc(f_j, Ens\_Cls) \tag{4.15}$$

where $EnsAcc(f_j, Ens\_Cls)$ reflects the ensemble accuracy of the already chosen classifiers RED_Chosen and the the candidate classifier f_j. It corresponds to the recognition rate computed following to equation 2.40.

To put it more simply, both *AD*-*AF* and the *EA*-*AF* assess the diversity of sets of attributes based on their similarity (i.e using similarity-based diversity measure (Algorithm 4.1)) and based on the outcomes of the resulting classifier (i.e using error based diversity measures presented in Section 1.2.1). With regard to the time complexity, these two approaches are more consuming in comparison with the *DR* one. They have $|RED|$ iterations for computing the diversity between the existing reducts and $|RED_{New}|$ iterations for the fitness function computation. In sum, they have a complexity equals $O(M \times |RED + RED_{New}|)$.

4.4 Experimentation settings and results

Following the same experimentation settings of chapter 3, we evaluate the performance of our three rough set based selective ensemble EEk-NN classifiers. Before the evaluation study, we present, in Table 4.2, the number of generated reducts for the different used databases. From this table, we can remark that a huge number of reducts have been generated. For instance, we have 8191 reducts for the Hepatitis databases and we have 975 reducts for the Thoracic Surger one. The obtained results prove the real need of reduct selection heuristics for ensemble classifiers. By the following, we evaluate our three reducts selection heuristics in terms of ensemble size and reduct diversity. One important element which has to be highlighted concerns the maximum number M of selected reducts, meaning selected classifiers. According to a study conducted by Opitz and Maclin (1999), ensembles of 25 classifiers are sufficient for improving the ensemble performance. Thus, for our experimentation, we set M to 25. Another key issue which has to be addressed is the number of neighbors yielding satisfactory results. Herein, we evaluate five values of k which respectively correspond to 1, 3, 5, 7 and 9.

4.4.1 Ensemble size

Herein, we evaluate our three proposed heuristics when relied on the ensemble size as an evaluation criterion. The obtained results are given from Table 4.3 to Table 4.8 where we can remark that both *AD*-*AF* and *EA*-*AF* methods have yielded smallest ensemble in compared with the the *DR* approach. Let us take

Table 4.2: Reduct generation for the different databases

	Feature subsets	Generated reducts
Heart	2^{13}	127
Japanese	2^{15}	511
Vote Records	2^{16}	163
Hepatitis	2^{19}	8191
Thoracic Surger	2^{13}	975
Wine	2^{16}	1824

the Hepatitis database with *k* equals 1 as example, the ensemble size achieved by respectively the *DR*, the *AD-AF* and the *EA-AF* are equal to 25, 4 and 4. From this point of view, we can deduce the efficiency of the *AD-AF* and *EA-AF* approaches for generating ensemble EE*k*-NN with reduced size.

Table 4.3: Ensemble size for Heart database

	DR	*AD-AF*	*EA-AF*
k=1	25	4	2
k=3	25	3	3
k=5	25	2	3
k=7	25	4	2
k=9	25	2	3

Table 4.4: Ensemble size for Japanese database

	DR	*AD-AF*	*EA-AF*
k=1	25	3	2
k=3	25	3	2
k=5	25	3	3
k=7	25	3	3
k=9	25	3	2

Table 4.5: Ensemble size for Vote Records database

	DR	*AD-AF*	*EA-AF*
k=1	25	3	3
k=3	25	3	2
k=5	25	3	2
k=7	25	3	3
k=9	25	3	2

Table 4.6: Ensemble size for Hepatitis database

	DR	*AD-AF*	*EA-AF*
k=1	25	4	4
k=3	25	5	3
k=5	25	4	4
k=7	25	3	3
k=9	25	4	4

Table 4.7: Ensemble size for Thoracic Surgery database

	DR	*AD-AF*	*EA-AF*
k=1	25	4	3
k=3	25	4	3
k=5	25	4	4
k=7	25	4	4
k=9	25	4	3

Table 4.8: Ensemble size for Wine database

	DR	*AD-AF*	*EA-AF*
k=1	25	3	3
k=3	25	4	3
k=5	25	4	3
k=7	25	3	3
k=9	25	3	3

4.4.2 Reduct diversity

Let us move on now to evaluate our proposed reduct selection approach when relied on the diversity between the different obtained reducts. We mainly relied on the Jaccard distance J_δ for measuring the reduct diversity. This measure highly depends on the number of reducts. In fact, the maximum diversity is yielded when there is an empty intersection of the generated reducts. It is set to:

$$J_\delta = \frac{|R_1 \cup R_2 \cup \ldots \cup R_{M'}| - |R_1 \cap R_2 \cap \ldots \cap R_{M'}|}{|R_1 \cup R_2 \cup \ldots \cup R_{M'}|} \tag{4.16}$$

The obtained results are given From Table 4.9 to Table 4.14. From these tables, we can remark that the *DR* method has achieved in the almots cases the most diverse reducts comparatively with the *AD-AF* and *EA-AF* approaches for almost databases. This can be explained by specific feature of the Jaccard measure. In fact, it promotes the ensemble constructed with the largest number of reducts. The results still show that the *AD-AF* and *EA-AF* methods are able to provide sets of reducts with higher diversity compared to *DR* on some dataset (e.g., Hepatitis) and approach the DR performance on others (e.g., Vote) and even be better under some conditions (e.g., Heart, for k equals 7 and k equals 9).

Table 4.9: Reduct diversity for Heart database

	DR	*AD-AF*	*EA-AF*
k=1	0.6	0.5	0.54
k=3	0.6	0.5	0.5
k=5	0.6	0.5	0.54
k=7	0.6	0.42	0.86
k=9	0.6	0.42	0.86

Table 4.10: Reduct diversity for Japanese database

	DR	*AD-AF*	*EA-AF*
k=1	0.69	0.65	0.65
k=3	0.69	0.67	0.69
k=5	0.69	0.65	0.63
k=7	0.69	0.69	0.64
k=9	0.69	0.66	0.69

Table 4.11: Reduct diversity for Vote Records database

	DR	*AD-AF*	*EA-AF*
k=1	0.5	0.48	0.49
k=3	0.5	0.49	0.47
k=5	0.5	0.5	0.48
k=7	0.5	0.48	0.50
k=9	0.5	0.5	0.48

Table 4.12: Reduct diversity for Hepatitis database

	DR	*AD-AF*	*EA-AF*
k=1	0.54	0.58	0.54
k=3	0.54	0.58	0.58
k=5	0.54	0.58	0.46
k=7	0.54	0.58	0.54
k=9	0.54	0.58	0.5

Table 4.13: Reduct diversity for Thoracic Surger database

	DR	*AD-AF*	*EA-AF*
k=1	0.94	0.92	0.90
k=3	0.94	0.93	0.94
k=5	0.94	0.93	0.92
k=7	0.94	0.92	0.90
k=9	0.94	0.93	0.91

Table 4.14: Reduct diversity for Wine Data database

	DR	*AD-AF*	*EA-AF*
k=1	0.82	0.79	0.79
k=3	0.82	0.75	0.81
k=5	0.82	0.79	0.76
k=7	0.82	0.81	0.74
k=9	0.82	0.77	0.76

4.5 Conclusion

We have propose, in this Chapter, a novel framework for selecting a successfully ensemble of the EE*k*-NN classifier for addressing data with evidential attributes. Our framework consists firstly of generating all possible reducts and then selecting the most suitable ones for an ensemble of EE*k*-NN classifiers. Three approaches have been proposed for selecting the best reducts, namely the Diversity Reduct method (*DR*), the Accuracy-Diversity Assessment Function method (*AD-AF*) and the Ensemble Accuracy Assessment Function method (*EA*-*AF*). These mentioned approaches have been compared in terms of the ensemble size and the reduct diversity. The achieved results have proven the efficiency of both the *AD*-*AF* and the *EA*-*AF* methods over the *DR* one according to the ensemble size, while the *DR* has yielded the most diverse reduct.

In the next Chapter, we select the reduct selection approaches yielding the best classification performance and we study the impact od some combination rules in the ensemble performance.

Chapter 5

Combining selective ensemble EEk-NNs

Contents

5.1 Introduction 93

5.2 Dempster's rule for merging classifiers 94

5.3 Mixed rule for combining classifiers 100

5.3.1 Classifier diversity 100

5.3.2 The optimized t-norm for evidential classifier combination 101

5.3.3 A comparative study 103

5.4 Conclusion 105

5.1 Introduction

The construction of successfully ensemble classifiers is still a hot undergoing research topic and relied mainly on classifier selection and classifier fusion. In the previous Chapter, we have proposed three classifier selection approaches and we have compared them in terms of ensemble size and the training set diversity. Unfortunately, both measures are insufficient and not suitable to decide the best classifier selection approach. Thus, in this Chapter, we evaluate and compare our three classifier selection methods using some standard information retrieval measures. The comparison with Random Selected Reducts (*RSR*), meaning random sampling with replacement, will also be conducted. The best chosen method will then be applied for identifying the most appropriate rule.

The random subspace method is similar to bagging except that the features ("attributes", "predictors", "independent variables") are randomly sampled, with replacement, for each learner

In this Chapter, we suppose firstly the independence between classifiers and we apply the Dempster operator for making decision about the most appropriate classifier selection technique. The choice of this rule is justified by its great ability to merge independent information. However, the classifier independence in an ensemble seems to be an unreliable assumption and an optimized t-norm rule with behavior ranging between the Dempster and the cautious rules has to be used as an alternative Quost et al. (2011). Thus, in this Chapter, we conduct a comparative study between the Dempster, the cautious and the optimized t-norm rules. The main aim behind this study is to identify the most appropriate combination rule for an ensemble classifiers.

5.2 Dempster's rule for merging classifiers

The belief function theory has not only the advantage to manage and represent uncertainty. It proposes also a set of combination rules to merge evidence acquired from several information sources, notably the evidential outputs of an ensemble of classifiers. The Dempster operator is the well used belief function rule in the context of classifier fusion within the evidence theory (Quost et al., 2011). From this, we use the Dempster rule for combining the selected individual based classifiers obtained by the RD, the AD-AF and the EA-AF methods. Following the same experimentation settings presented in Chapter 3, we carry out a comparative study between these classifier selection methods when reling on the PCC, the recall and the precision as assessment measures. The experimentation results for the different databases are given from Table 5.1 to Table 5.6. Plots showing the general tendency of the classifiers performance across the different datasets with k equals 7 are given from Figure 5.1 to Figure 5.3. From the obtained redults, we can remarked that the performance of an ensemble system is greatly influenced by the selected reduct approach.

Ensembles built from random selected reducts (*RSR*) provide the most poorly performance comparatively with the *DR*, the *AD-AF* and the *EA-AF* approaches for all the tested databases and for all values of *k*. Taken as an example the Heart dataset with *k* equals 1, the PCC results for respectively the *RSR*, *DR*, the *AD-AF* and the *EA-AF* are equal to 55.88 %, 70%, 73.70% and 77.77%. The recall results are equal to 61.70%, 70.19%,74.02% and 77.68%. The precision results are equal to 60.56%, 69.74%, 73.03% and 77.32%. This conclusion may be justified by the fact that redundant feature may be part of the selected reducts.

Table 5.1: Combination results for Heart database

		RSR	*DR*	*AD-AF*	*AE-AF*
k=1	PCC	55.88 ± 0.07	70.00 ± 0.04	73.70 ± 1.16	**77.77 ± 1.23**
	Recall	61.70 ± 0.07	70.19 ± 0.04	74.02 ± 1.16	**77.68 ± 1.23**
	Precision	60.56 ± 0.06	69.74 ± 0.03	73.03 ± 1.15	**77.32 ± 1.22**
k=3	PCC	61.48 ± 0.03	75.92 ± 0.02	75.92 ± 1.18	**77.77 ± 1.25**
	Recall	67.90 ± 0.1	75.82 ± 0.02	75.99 ± 1.19	**77.77 ± 1.25**
	Precision	57.49 ± 0.03	75.64 ± 0.02	75.39 ± 1.18	**77.52 ± 1.24**
k=5	PCC	72.96 ± 0.06	78.14 ± 0.04	**78.88 ± 1.24**	**78.88 ± 1.25**
	Recall	73.84 ± 0.05	78.16 ± 0.04	78.82 ± 1.24	**78.93 ± 1.25**
	Precision	71.70 ± 0.06	77.59 ± 0.04	78.53 ± 1.23	**78.59 ± 1.24**
k=7	PCC	72.96 ± 0.03	78.88 ± 0.03	78.88 ± 1.24	**82.86 ± 1.32**
	Recall	74.74 ± 0.02	79.32 ± 0.03	78.98 ± 1.24	**83.34 ± 1.33**
	Precision	71.13 ± 0.03	78.27 ± 0.03	78.62 ± 1.24	**82.43 ± 1.33**
k=9	PCC	74.44 ± 0.03	78.14 ± 0.03	80.00 ± 1.26	**82.59 ± 1.30**
	Recall	75.95 ± 0.03	78.67 ± 0.03	80.31 ± 1.27	**82.90 ± 1.30**
	Precision	73.00 ± 0.01	77.43 ± 0.03	79.26 ± 1.25	**81.95 ± 1.29**

Table 5.2: Combination results for Japanese database

		RSR	*DR*	*AD-AF*	*AE-AF*
k=1	PCC	55.94 ± 0.02	68.98 ±0.03	69.85 ± 1.11	**72.31 ± 1.16**
	Recall	62.90 ± 0.07	62.90 ± 0.07	63.02 ± 0.9	**64.48 ± 1.02**
	Precision	60.08 ± 0.09	66.95 ± 0.1	66.87 ± 1.06	**69.00 ± 1.10**
k=3	PCC	45.21 ± 0.3	73.76 ± 0.03	76.37 ± 1.25	**78.69 ± 1.30**
	Recall	56.22 ± 0.29	65.99 ± 0.09	67.22 ± 1.07	**68.68 ± 1.10**
	Precision	71.38 ± 0.26	71.10 ± 0.12	72.05 ± 1.16	**73.79 ± 1.20**
k=5	PCC	52.17 ± 0.26	71.59 ± 0.04	78.40 ± 1.28	**80.86 ± 1.33**
	Recall	65.83 ± 0.09	64.95 ± 0.08	69.26 ± 1.10	**71.00 ± 1.13**
	Precision	57.33 ± 0.05	71.81 ± 1.01	74.27 ± 1.19	**76.04 ± 1.23**
k=7	PCC	55.94 ± 0.26	70.86 ± 0.09	79 42 ± 1.31	**80.00 ± 1.30**
	Recall	63.72 ± 0.14	65.65 ± 0.1	69.19 ± 1.10	**69.90 ± 1.10**
	Precision	61.05 ± 1.12	68.55 ± 0.16	73.33 ± 1.18	**74.04 ± 1.18**
k=9	PCC	44.92 ± 0.3	70.43 ± 0.69	**81.01 ± 1.32**	**81.01 ± 1.32**
	Recall	65.62 ± 0.08	64.08 ± 0.09	**70.48 ± 1.11**	**70.48 ± 1.11**
	Precision	61.04 ± 0.21	66.04 ± 0.17	**74.81 ± 1.18**	**74.81 ± 1.18**

To cope with redundancy, we have proposed firstly the *DR* approach allowing the selection of diverse reducts from the original pool. The experimentation results have proven the impact of diversity when constructing an ensemble system. In fact, the PCC, the recall and the precision results achieved by the *DR* approach are greater than those obtained through the *RSR* method for almost all cases.

Table 5.3: Combination results for Vote Record database

		RSR	*DR*	*AD-AF*	*AE-AF*
k=1	PCC	92.87 ± 0.04	94.02 ± 0.04	94.35 ± 0.02	**95.17** ± **0.12**
	Recall	92.90 ± 0.04	93.92 ± 0.02	94.07 ± 0.03	**94.33** ± **0.2**
	Precision	92.21 ± 0.23	93.50 ± 0.03	94.45 ± 0.07	**95.02** ± **0.1**
k=3	PCC	94.02 ± 0.04	94.02 ± 0.03	94.22 ± 0.1	**94.46** ± **0.07**
	Recall	94.26 ± 0.04	94.45 ± 0.04	94.63 ± 0.02	**95.02** ± **0.1**
	Precision	93.15 ± 0.04	93.01 ± 0.04	94.25 ± 0.06	**94.63** ± **0.12**
k=5	PCC	93.33 ± 0.03	93.79 ± 0.03	94.01 ± 0.05	**94.22** ±**0.17**
	Recall	93.65 ± 0.04	93.79 ± 0.03	94.15 ± 0.06	**94.56** ± **0.13**
	Precision	92.34 ± 0.04	92.83 ± 0.04	93.17 ± 0.03	**93.76** ± **0.26**
k=7	PCC	93.79 ± 0.03	93.56 ± 0.03	94.22 ±0.12	**94.35** ± **0.17**
	Recall	94.13 ± 0.03	93.82 ± 0.03	94.25 ±0.07	**94.88** ± **0.27**
	Precision	92.83 ± 0.03	92.64 ± 0.04	92.94 ± 0.2	**93.56** ± **0.23**
k=9	PCC	93.16 ± 0.03	93.79 ± 0.03	94.45 ± 0.24	**94.63** ± **0.23**
	Recall	93.85 ± 0.03	94.02 ± 0.03	94.27 ± 0.03	**94.53** ± **0.24**
	Precision	92.64 ± 0.03	92.90 ± 0.04	93.19 ± 0.15	**93.76** ± **0.07**

Table 5.4: Combination results for Hepatitis database

		RSR	*DR*	*AD-AF*	*AE-AF*
k=1	PCC	45.16 ± 0.12	49.03 ± 0.1	49.23 ±0.06	**49.54** ±**0.13**
	Recall	78.46 ± 0.24	85.75 ±0.2	85.89 ± 0.23	**86.01** ±**0.12**
	Precision	54.03 ±0.10	49.71 ± 0.16	49.89 ± 0.25	**50.03** ± **0.1**
k=3	PCC	45.16 ± 0.13	47.09 ± 0.16	48.46 ± 0.23	**49.46** ± **0.17**
	Recall	70.05 ± 0.24	85.28 ± 0.19	85.96 ± 0.12	**86.25** ± **0.23**
	Precision	48.73 ± 0.12	48.64 ± 0.1	49.12 ±0.03	**49.22** ±**0.24**
k=5	PCC	43.87 ± 0.13	49.67 ± 0.13	50.22 ± 0.2	**50.34** ± 0.07
	Recall	71.48 ± 0.23	70.06 ± 0.02	70.35 ±0.14	**70.62** ± **0.23**
	Precision	58.60 ± 0.12	60.95 ± 0.03	62.09 ± 0.12	**63.47** ± **0.01**
k=7	PCC	50.32 ± 0.12	50.32 ± 0.16	51.02 ± 0.22	**51.45** ± **0.05**
	Recall	72.50 ± 0.22	71.33 ±0.25	73.45 ± 0.05	**74.62** ± **0.14**
	Precision	60.23 ± 0.13	60.57 ± 0.26	61.82 ± 0.11	**62.45** ± **0.27**
k=9	PCC	47.74 ± 0.13	49.67 ± 0.17	50.04 ± 0.14	**52.23** ± **0.22**
	Recall	71.39 ± 0.23	71.39 ± 0.23	72.45 ± 0.06	**73.12** ± **0.02**
	Precision	60.13 ± 0.14	59.99 ± 0.16	61.02 ± 0.17	**62.45** ± **0.29**

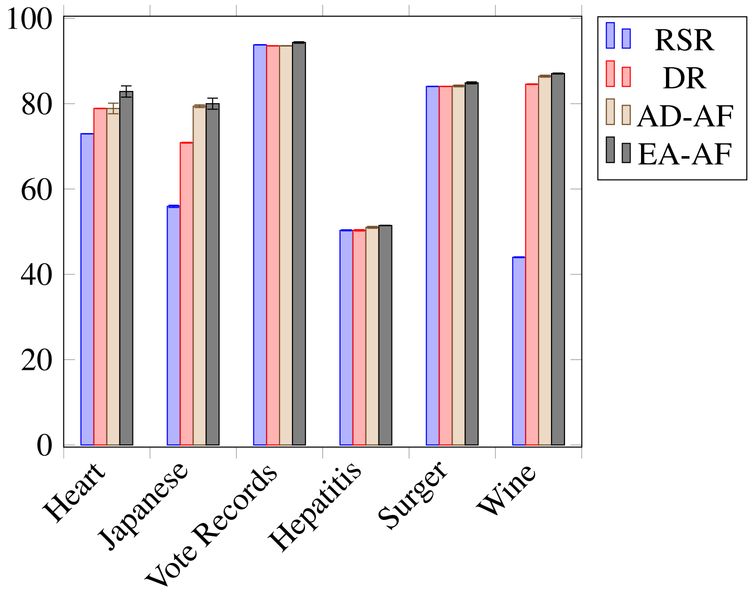


Figure 5.1: PCC results for *k* equals 7

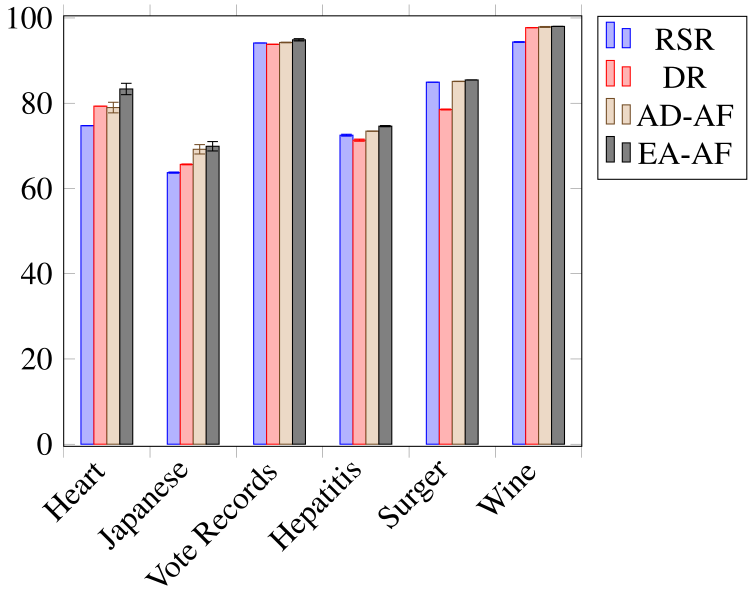


Figure 5.2: Recall results for *k* equals 7

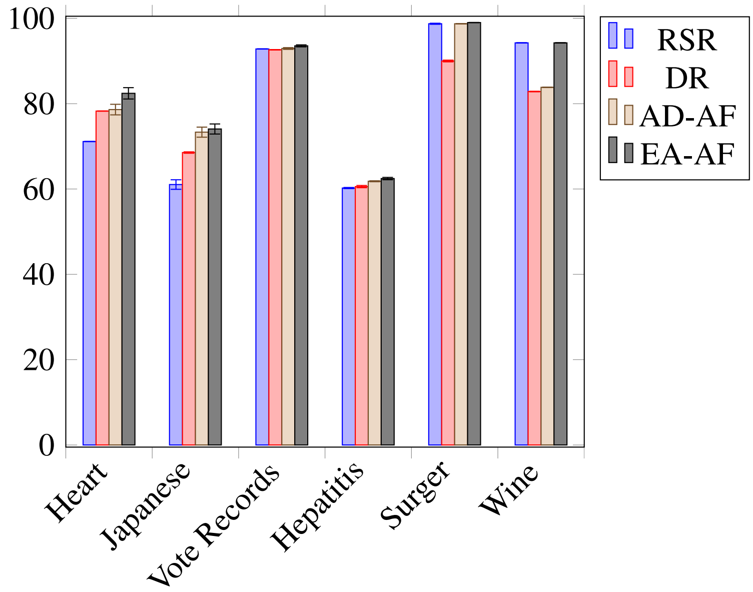


Figure 5.3: Precision results for k equals 7

Considering diversity between reducts is not only sufficient for achieving the best performance. Thus, we have proposed the *AD-AF* method that takes into consideration in addition to the diversity between the selected reducts the diversity and the accuracy of base classifiers. The obtained results, have proven the effectiveness of this approach over the *DR* one for almost all databases. For instance, in Japanese database with k equals 3, the PCC of the *DR* and the *AD-AF* techniques are equal respectively to 73.76 % and 76.37%, the recall values are equal to 65.99% and 67.22% and the precision values are equal to 71.10 % and 72.05%. Therefore, we can deduce the effectiveness of the *AD-AF* technique over the *DR* in terms of the ensemble size as well as the classification performance.

Table 5.5: Combination results for Wine database

		RSR	*DR*	*AD-AF*	*AE-AF*
k=1	PCC	56.57 ± 0.02	90.28 ± 0.04	90.45 ± 0.02	**91.15** ± **0.07**
	Recall	93.99 ± 0.08	98.49 ± 0.02	98.45 ± 0.23	**98.79** ± 0.02
	Precision	56.09 ± 0.02	88.41 ± 0.03	87.98 ± 0.23	**88.23** ± **0.01**
k=3	PCC	56.00 ± 0.03	90.28 ± 0.03	91.46 ±0.12	**92.52** ± **0.23**
	Recall	95.44 ± 0.06	98.85 ± 0.01	99.02 ± 0.03	**99.15** ± **0.03**
	Precision	54.33 ± 0.29	88.41 ± 0.02	88.62 ± 0.17	**88.91** ± **0.14**
k=5	PCC	52.00 ± 0.2	87.42 ± 0.05	88.12 ± 0.02	**91.22** ± **0.07**
	Recall	94.96 ± 0.06	98.59 ± 0.01	97.98 ± 0.12	**98.87** ± **0.03**
	Precision	50.23 ± 0.19	85.55 ± 0.03	87.26 ± 0.09	**89.54** ± **0.11**
k=7	PCC	44.00 ± 0.1	84.57 ± 0.08	86.45 ± 0.23	**87.07** ± **0.12**
	Recall	94.36 ± 0.09	97.73 ± 0.03	97.89 ± 0.11	**98.02** ± **0.01**
	Precision	41.46 ± 0.07	82.85 ± 0.06	83.84 ± 0.04	**94.26** ± **0.04**
k=9	PCC	40.00 ± 0.09	84.57 ± 0.08	85.22 ± 0.01	**86.17** ± **0.01**
	Recall	87.00 ± 0.2	97.73 ± 0.03	97.82 ± 0.01	**98.05** ± **0.01**
	Precision	40.95 ± 0.09	82.85 ± 0.07	83.14 ± 0.12	**84.29** ± **0.07**

Table 5.6: Combination results for Thoracic Surgery database

		RSR	*DR*	*AD-AF*	*AE-AF*
k=1	PCC	80.21 ± 0.03	77.23 ± 0.03	81.22 ± 0.02	**81.44** ± **0.05**
	Recall	53.13 ± 0.19	55.53 ± 0.06	61.18 ± 0.11	**61.48** ± **0.12**
	Precision	60.79 ± 0.17	53.88 ± 0.05	60.88 ± 0.02	**61.22** ± **0.23**
k=3	PCC	82.55 ± 0.03	82.55 ± 0.02	83.16 ± 0.01	**83.23** ± **0.11**
	Recall	62.25 ± 0.12	84.74 ± 0.03	85.25 ± 0.03	**85.49** ±**0.07**
	Precision	60.93 ± 0.2	97.03 ± 0.01	97.63 ± 0.11	**97.89** ± **0.11**
k=5	PCC	84.04 ± 0.02	83.61 ± 0.02	84.16 ± 0.03	**84.46** ± **0.01**
	Recall	75.04 ± 0.14	74.23 ± 0.13	76.25 ± 0.02	**76.67** ± **0.02**
	Precision	80.39 ± 0.25	80.34 ± 0.25	80.77 ± 0.01	**81.10** ± **0.01**
k=7	PCC	84.04 ± 0.03	84.04 ± 0.02	84.16 ± 0.2	**84.89** ± **0.23**
	Recall	84.94 ± 0.03	78.53 ± 0.11	85.12 ± 0.03	**85.46** ± **0.03**
	Precision	98.74 ± 0.12	90.01 ± 0.20	98.74 ± 0.03	**99.02** ± **0.03**
k=9	PCC	84.89 ±	83.40 ± 0.62	85.15 ± 0.02	**96.68** ± **0.02**
	Recall	85.07 ± 0.03	84.88 ± 0.03	85.13 ± 0.07	**85.44** ± **0.11**
	Precision	99.75 ± 0.05	98.05 ± 0.01	99.83 ± 0.01	**99.94** ± **0.01**

In comparison with the *AD-AF* technique, our third reduct selection approach, refereed to as *EA-AF*, consists of selecting reducts when taking into consideration the accuracy of the ensemble instead of the accuracy and the diversity of individual classifiers. As can be remarked from Table 5.1 to Table 5.6, the *EA-AF* has yielded the best classification performance for all the tested databases and for the different values of *k*. From an experimental point of view, a classifier construction strategy relying on the diversity between reducts and the accuracy of an ensemble system is well suited procedure for yielding good ensemble performance.

5.3 Mixed rule for combining classifiers

One main point to be addressed is that Dempster'rule assumes evidence to be fully independent. As remarked From Table 4.9 to Table 4.14, the obtained reducts through the *EA-AF* technique are low correlated. For that reason, the Dempster rule is not very well suited for merging classifier outputs. Alternatively, the optimized t-norm based rule has been developed for addressing the case of both dependent and independent classifiers thanks to its behavior ranging between the Dempster rule and the cautious rule (Quost et al., 2011). By the following, we compute firstly the degree of disagreement between the base classifiers using the *EA-AF* approach. Then, we briefly review the basic concepts of the optimized t-norm based rule. Subsequently, we carry out a comparative study between the Dempster, the cautious and the optimized t-norm combination rules. The underling idea throughout this study is to pick out the most adequate combination rule for evidential classifier ensemble.

5.3.1 Classifier diversity

As stated in the beginning of this dissertation, several diversity measures have been reported in the literature and the choice of the most convenient one is still an open question. Herein, we relied on Equation 1.4 as a measure a diversity for computing the independence between the individual classifiers. The degree of independence between classifiers are given in Table 5.7.

Table 5.7: The degree of independence between individual classifiers

	Heart	Japanese	Vote Records	Hepatitis	Wine	Thoracic Surgery
k=1	0.34 ± 0.06	0.35 ± 0.05	0.42 ± 0.07	0.22 ± 0.03	0.31 ± 0.23	0.35 ± 0.05
k=3	0.30 ± 0.09	0.29 ± 0.05	0.39 ± 0.01	0.25 ± 0.09	0.31 ± 0.13	0.33 ± 0.07
k=5	0.33 ± 0.07	0.32 ± 0.03	0.41 ± 0.1	0.21 ± 0.01	0.29 ± 0.07	0.37 ± 0.01
k=7	0.25 ± 0.08	0.30 ± 0.06	0.42 ± 0.07	0.23 ± 0.05	0.28 ± 0.03	0.34 ± 0.03
k=9	0.26 ± 0.05	0.27 ± 0.06	0.38 ± 0.05	0.22 ± 0.05	0.32 ± 0.15	0.31 ± 0.02

According to Table 5.7, we can conclude that the individual classifiers are not fully in disagreement. In this context, the Dempster rule is not well suited and the optimized t-norm based operator can be applied as an alternative (Quost et al., 2011). By the following, we describe in more details this rule.

5.3.2 The optimized t-norm for evidential classifier combination

Restricting to separable mass function [1], Quost et al. (2011) have relied on the Frank t-norm family for processing both the Dempster and the cautious combination rules as special cases. It is set to:

$$xT_s y = log_s(1 + \frac{(s^x - 1)(s^y - 1)}{s - 1}) \tag{5.1}$$

where log_s reflects the logarithm function with base s>0. Noting that each value of the parameter s defines a t-norm and a combination rule as follows:

$$m_1 \textcircled{T}_s m_2 = \textcircled{\cap}_{A \subset \Theta} A^{w_1(A) T_s w_2(A)} \tag{5.2}$$

Assume that m_1 and m_2 are separable mass functions provided by evidential classifiers. If classifiers are fully independent, the $\textcircled{T}_0$ operator with s equals 0 corresponds to the Dempster operator. In case of not fully independence, t-norm based

---

[1] A mass function is separable if it can be written as the combination of simple mass functions.

rules (i.e when the parameter s is in the range $]0,1]$) including the cautious one may achieve better performance. The issue of optimizing the t-norm based rule for yielding best performance has already been answered (Quost et al., 2011). Two strategies have been investigated. The first one consists of learning a single rule by minimizing an error criterion. Assume that M is the number of classifiers to be combined, H is the number of validation sets and E^{s_h} is the classification error achieved by optimizing the parameter s_h of the t-norm rule on each validation set $h \in H$. The optimal parameter $\hat{s}$ is that satisfying the following function.

$$\hat{s} = arg \min_{0<s\leq 1} \frac{\sum E^{s_h}}{H} \tag{5.3}$$

Regarding the second strategy, it consists of a two-step procedure. The first step is to group classifiers in terms of their dependencies. A within-cluster rule has to be used for fusing dependent classifiers within every cluster, while a between-cluster rule has to be used for combining the outputs provided by each cluster. To put it more clearly, the optmized t-norm strategy is depicted in Figure 5.4 for the case of 6 classifiers C=$\{C_1 \ldots C_6\}$ grouped into two clusters with size equals 3.

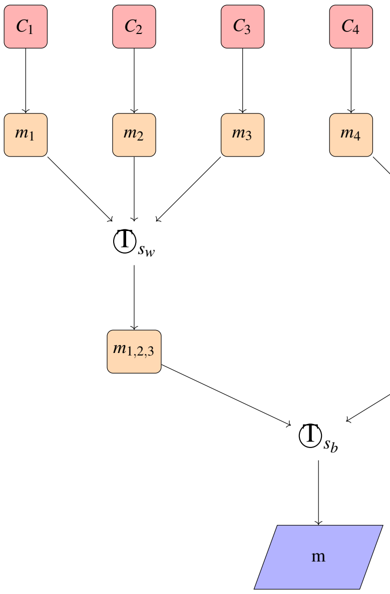


Figure 5.4: The optimized t-norm based rule strategy

The clustering process may take different forms. In an effective way, it consists of increasing the diversity between groups, while decreasing the diversity inside each group. Two steps should be followed for identifying clusters. The first one consists of calculating diversity between classifiers. Quost et al. (2011) have proposed two ways for diversity computing. The first way concerns the use of a pairwise measure (Kuncheva & Whitaker, 2003), particularly the disagreement one. The second way is to use the Jousselme distance (A. Jousselme et al., 2001) for measuring the disagreement between classifier outputs. The distance between classifiers has to be computed as the average distance yielded through all training instances. Regarding the second phase, is consists of determining clusters when relied on the hierarchical clustering algorithm (Dubes & Jain, 1980). This clustering technique has the advantage to determine the clusters' number within a fairly straightforward manner. The conducted experimentations have proven that both the Jousselme distance and the disagreement measure achieve almost identical results.

Once the clustering process has been made, classifiers have to be merged following two levels. The first level consists of combing classifiers within each cluster through the within-cluster rule with a parameter value s_w and the second level is to merge clusters' outputs using the between-cluster rule with a parameter value s_b. The idea behind the optimized t-norm rule is to estimate the pair of values $(\widehat{s_w},\widehat{s_b})$ minimizing the cross validation error over H validation sets. That is mean, with each of the $h \in H$ subsamples, values $a_1 \approx 0, \ldots a_R \approx 1$ equally spaced on a logarithmic scale are picking out as candidate values for s_b. Assume that a_i is the selected s_b parameter value, values in the range $[a_1, \ldots, a_i]$ are regarded as candidate values for s_w. The pair $(\widehat{s_w},\widehat{s_b})$ minimizing the average error over the H validation sets has to be retained. In sum, we perform $R \times (R - 1)/2$ evaluation for each subsample h.

5.3.3 A comparative study

As indicated previously, Dempster's rule is not the most adequate operator for combining decisions yielded by a classifier ensemble. Therefore, in this subsection, we investigate the impact of combination rules on the ensemble performance. From the plethora of rules that exist, we only consider Dempster's rule, the cautious rule and the optimized t-norm based rule. A comparison with single EE*k*-NN classifier for $k \in \{1,3,5,7,9\}$ is also be considered. Assuming that we follow a 5-fold cross validation, the comparative results in terms of the PCC criterion are presented in Table 5.8 and a plot for a value of k equals 7 is given in Figure 5.5

Table 5.8: Combination results: PCC

		Heart	Japanese	Vote Records	Hepatitis	Wine	Thoracic Surgery	Average
k=1	Dempster	77.77 ± 1.23	72.31 ± 1.16	95.17 ± 0.12	49.54 ± 0.13	91.15 ± 0.07	81.44 ± 0.05	77.89
	Cautious	76.66 ± 1.23	71.14 ± 0.12	94.23 ± 0.32	49.32 ± 0.12	82.79 ±0.03	82.21 ± 0.03	77.22
	Optimized T-norm	**83.70 ± 1.34**	72.31 ± 1.16	**95.69 ± 0.21**	**52.24 ± 0.17**	**94.32 ± 0.17**	**83.63 ± 1.12**	**80.31**
	EE1-NN	45.58 ± 0.04	**81.15 ± 0.07**	90.57 ± 0.03	50.32 ± 0.14	88.57 ± 0.09	78.29 ± 0.03	72.45
k=3	Dempster	77.77 ±1.25	78.69 ± 1.30	94.46 ± 0.07	49.46 ±0.17	92.52 ± 0.23	83.23 ± 0.11	79.35
	Cautious	80.00 ± 1.26	78.69 ±1.10	94.78 ± 0.03	49.98 ± 0.12	93.12 ± 0.07	83.67 ± 0.11	80.04
	Optimized T-norm	**85.92 ± 1.38**	78.84 ±1.30	**95.08 ± 1.12**	**52.14 ± 0.09**	**95.42 ± 1.15**	**82.04 ± 1.13**	**81.90**
	EE3-NN	64.07 ± 0.05	**83.62 ± 0.07**	93.33 ± 0.04	50.32 ± 0.14	84.00 ±0.06	81.27 ± 0.03	76.10
k=5	Dempster	78.88 ±1.25	80.86 ±1.33	94.22 ± 0.17	50.34 ± 0.07	91.22 ± 0.07	84.46 ± 0.01	79.99
	Cautious	81.48 ± 1.30	81.5 ± 1.12	94.34 ± 0.23	51.56 ± 1.04	93.37 ± 0.11	84.82 ± 0.05	81.02
	Optimized T-norm	85.55 ± 1.35	81.30 ±1.30	**94.68 ± 0.13**	**53.42 ± 1.16**	**95.42 ± 1.12**	**85.14 ± 0.11**	**82.58**
	EE5-NN	77.40 ± 0.03	**83.91 ±0.07**	93.56 ± 0.03	49.63 ± 0.11	85.14 ± 0.06	81.91 ±0.03	78.49
k=7	Dempster	82.96 ± 1.32	80.00 ± 1.3	94.35 ± 0.17	51.45 ± 0.15	87.07 ± 0.12	84.89 ± 0.23	80.12
	Cautious	83.70 ± 1.33	80.54 ± 1.13	94.78 ± 0.13	52 .27 ± 0.22	91.18 ± 0.04	85.11 ± 0.12	81.26
	Optimized T-norm	**86.29 ± 1.37**	81.01 ± 1.33	**94.98 ± 0.23**	**54.18 ± 1.23**	**93.71 ± 1.49**	**85.66 ± 0.17**	**82.63**
	EE7-NN	81.11 ± 0.05	**84.20 ± 0.08**	93.56 ± 0.04	52.25 ± 0.13	85.14 ± 0.03	82.55 ± 0.02	79.80
k=9	Dempster	82.59 ±1.30	81.01 ± 1.32	94.63 ± 0.23	52.23 ± 0.22	86.17 ± 0.01	96.68 ± 0.02	82.21
	Cautious	84.44 ± 1.32	81.64 ± 1.23	94.63 ± 0.23	49.54 ± 0.13	86.89 ± 1.13	97.02 ± 0.02	82.44
	Optimized T-norm	**86.66 ± 1.39**	82.02 ± 1.34	**95.68 ± 1.31**	**54.48 ± 0.15**	**87.23 ± 1.27**	**98.14 ± 0.12**	**84.03**
	EE9-NN	81.85 ± 0.05	**85.21 ± 0.06**	92.64 ± 0.05	50.32 ± 0.11	88.00 ± 0.03	82.97 ± 0.02	80.17

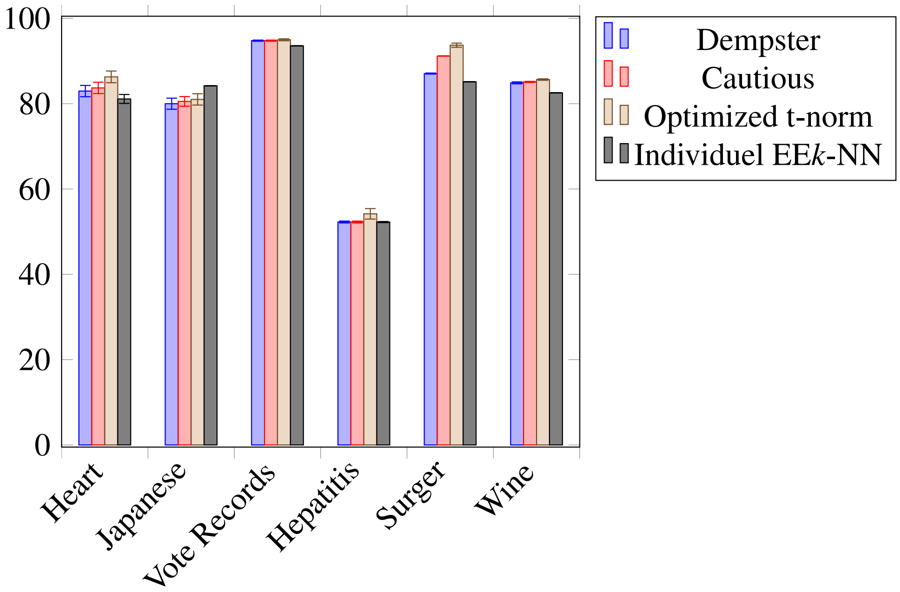


Figure 5.5: A comparison of the PCC results for a value of k equals 7

From these studied rules, the Demspter one has the worst performance comparatively to the t-norm optimized and the cautious rules for almost all cases. This result sustains the conclusion of Quost et al. (2011) about the independence of the individual classifiers. On the other hand, the optimized t-norm based rule gives the best results and it has also outperformed the results yielded by the individual classifier for all the databases, except the Japanese one. These results could be explained by the mathematical behavior of this rule. Indeed, it takes into consideration the dependence and the independence between individual classifiers.

One important point to be addressed is that ensemble classifier systems are not suitable for all cases (e.g the Japanese database). However, individual classifiers may perform well especially for databases with relevant and non redundant attributes.

5.4 Conclusion

In this Chapter, we made experiments to investigate the impact feature subspaces on the ensemble classifier performance. The obtained results have proven the effectiveness and the efficiency of the *EA-AF* approach. In fact, it allows to construct smallest ensembles that are built from high diverse reducts. We have also discussed the impact of three well-known combination operators on the classification performance. We have experimentally proven the efficiency of the optimized t-norm rule comparatively with the the Dempster and the cautious ones. This is may be justified by its ability to consider the independence factor when making the fusion process.

Conclusion

Ensemble classifier is regarded as a successful way for solving several machine learning problems. Since the construction of a good ensemble is a vital necessity, numerous heuristics have been introduced for several years now. One among them is rough set based ensemble classifier. Despite its obvious importance, evidential data have not been considered by a such ensemble classifier strategy.

The main aim underlying our dissertation is to develop a novel rough set based ensemble classifier for processing evidential data. Initially, we have suggested three newest evidential machine learning approaches. Notably, we have proposed two decision tree classifiers that differ on the way of decision attribute selection (Trabelsi, Elouedi, & Lefèvre, 2016c, 2016b). The thirs proposed algorithm is a *k*-NN approach called Evidential Editing *k*-NN classifier (Trabelsi, Elouedi, & Lefèvre, 2017). A comparative study between these algorithms has been carried out. The yielded results have proven the efficiency of the *k*-NN version over the proposed decision trees. This concern is justified by a simple theoretical consideration: the splitting process within our proposed generated decision tree induces trees with several branches. This may significantly increase the chance of overfitting and affect the classification performance. For that reason, we have proposed to construct an rough set based ensemble EE*k*-NN classifiers (Trabelsi, Elouedi, & Lefèvre, 2017, 2018).

As the standard ensemble classifier approach, the construction of an ensemble system follows two main steps: the selection of base individual classifiers (i.e meaning the selection of the most suitable reducts) and the choice of the most appropriate fusion rule.

Our individual classifier selection process is primarily based on the feature subspaces (reducts) selection procedure. To sum up, the choice of the best subsets of features requires two steps: the generation of all possible reducts and the selection of the most appropriate ones. We have proposed a novel algorithm for extracting all possible reducts within an evidential context. Our proposed approach for reduct selection is an extension of the SAVGenetic reducer, a Rosetta toolkit for generating all possible reduct (El-Monsef et al., 2003). In analogy to SAVGenetic approach, our proposed algorithm starts by computing a discernibility matrix from a given evidential data. Then, the non empty set of the obtained matrix have to be used for picking out approximate hitting sets, meaning approximate reducts.

Regarding the choice of the most suitable reducts, we have proposed three approaches, namely Diversity Reduct (*DR*), Accuracy-Diversity Assessment Function (*AD-AF*) and Ensemble Accuracy Assessment Function (*EA-AF*).

Let *M* be the maximum size of ensemble classifiers to be constructed. Our *DR* approach consists of selecting at most *M* diverse reducts among the pool of all reducts. As regards the *AD-AF* technique, it allows to select at most *M* diverse reducts maximizing a fitness function defined by an adaptive weighting between the accuracy and the diversity of the individual base classifiers. The latter approach (*EA-AF*) consists of choosing at most *M* diverse reducts allowing the construction of classifier ensemble with the highest ensemble accuracy. The main aim behind these three proposed methods is to evaluate the impact of the feature subsets on the ensemble performance.

For the evaluation process, we have supposed the independence between the base individual classifiers and we have relied on the Dempster operator for merge classifiers. This hypothesis has been formulated because it comes to merge individual classifiers trained with diverse reducts. The obtained results have proven the efficiency of the *EA-AF* solution comparatively with the *DR* and the *AD-AF* ones. In fact, it allows to generate ensemble classifiers with smallest size and highest performance.

However, a more elaborated study has proven the non totally independence between the constructed ensemble classifiers. This comes back to the fact that the selected reducts has some features in common. Accordingly, the Dempster operator may not be the most suitable fusion rule. Alternatively, the optimized t-norm based rule has been introduced to cope with independent and non fully independent classifiers. In fact, it has a behavior ranging between the Dempster rule (i.e for the case of fully independent classifiers) and the cautious rule (i.e for the case of not fully independent classifiers) (Quost et al., 2011). A comparative study between the optimized t-norm, the Dempster and the cautious rules has been carried out and the obtained results have proven the efficiency of the optimized t-norm based rule over the two other rules.

Regarding the future research directions, we look forward investigating more robust techniques for uncertain data modeling within the belief function framework. Notably, the case of incomplete data should be well studied. In fact, missigness representation may affect the prediction results (A.-L. Jousselme & Maupin, 2013).

It seems interesting to propose evidential decision tree pruning methods and to develop rough set based forest techniques. The performance of random forests has to be compared with rough set based ensemble EEk-NN.

We opt also to determine new research leads for all possible reduct generation when optimizing the configuration settings. In fact, our actually proposed solution depends on the choice of the tolerance threshold T that can affect the ensemble performance.

Concerning the fusion level, we have proven in a previous work the efficiency of the CWAC rule in comparison with the Dempster and the conjunctive rules(Trabelsi, Elouedi, & Lefèvre, 2015a). On the other side, we have experimentally proven the efficiency of the cautious CWAC rule in comparison with the cautious conjunctive operator and its normalized version (Trabelsi, Elouedi, & Lefèvre, 2015b). Thus, it could be useful to study the behavior of a combination rule with an adaptive weighting between the CWAC rule and the cautious CWAC rule.

Author publications

- **Journal paper:**
  - Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2016). Comparing dependent combination rules under the belief classifier fusion framework. Soft Computing, 1-14. (Impact Factor 2016 : 2.472)
- **Journal paper under revision:**
  - Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2018). Decision tree classifiers form totally uncertain data. Fuzzy Sets and Systems. (Impact Factor 2018 : 2.37)
- **International conferences:**
  - Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2018). Ensemble Enhanced Evidential *k*-NN classifier through rough set reducts. In proceedings of 17th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference (IPMU) (pp. 383–394).
  - Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2018). Ensemble Evidential Editing *k*-NNs through rough set reducts. In proceedings of 13th International FLINS conference on Data Science and Knowledge Engineering for Sensing Decision Support (FLINS).
  - Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2017). Ensemble Enhanced Evidential k-NN classifier through random subspaces. In proceedings of 10th European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQUARU) (pp. 212-221).

  - Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2017). A novel k-NN approach for data with uncertain attribute values. In proceedings of 30th International Conference on Industrial, Engineering, Other Applications of Applied Intelligent Systems (IEA/AIE) (pp. 57-64).
  - Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2016). Handling uncertain attribute values in decision tree classifier using the belief function theory. In proceedings of 17th International Conference on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA) (pp. 26-35).
  - Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2016). Feature selection from partially uncertain data within the belief function framework. In proceedings of 16th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU) (pp. 643-655).
  - Trabelsi, A., Elouedi, Z.,& Lefèvre, E. (2015). Belief function combination: Comparative study within the classifier fusion framework. In proceedings of 1st International Conference on Advanced Intelligent System and Informatics (AISI) (pp. 425-435).
  - Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2015). Classifier fusion within the belief function framework using dependent combination rules. In proceedings of 22nd International Symposium on Methodologies for Intelligent Systems (ISMIS) (pp. 133-138).

- **National conference:**
  - Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2016). New decision tree classifier for dealing with partially uncertain data. In proceedings of 25ème Rencontres francophones sur la Logique Floue et ses Applications (LFA) (pp. 57-64).

References

Al-Ani, A., & Deriche, M. (2002). A new technique for combining multiple classifiers using the dempster-shafer theory of evidence. *Journal of Artificial Intelligence Research*, 333–361.

Appriou, A. (1999). Multisensor signal processing in the framework of the theory of evidence. *Application of Mathematical Signal Processing Techniques to Mission Systems*, 1-5.

Barnett, J. A. (1991). Calculating Dempster–Shafer plausibility. *IEEE transactions on Pattern Analysis and Machine Intelligence*, *13*(6), 599–602.

Bell, D. A., & Wang, H. (2000). A formalism for relevance and its application in feature subset selection. *Machine learning*, *41*(2), 175–195.

Bhattacharjee, D., Basu, D. K., Nasipuri, M., & Kundu, M. (2010). Reduction of feature vectors using rough set theory for human face recognition. *arXiv preprint arXiv:1005.4044*.

Bloch, I. (1996). Information combination operators for data fusion: a comparative review with classification. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, *26*(1), 52-67.

Boubaker, J., Elouedi, Z., & Lefèvre, E. (2013). Conflict management with dependent information sources in the belief function framework. In *proceedings of the 14th international symposium on computational intelligence and informatics* (pp. 393–398).

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth.

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Information Fusion*, *6*(1), 5–20.

Bryll, R., Gutierrez-Osuna, R., & Quek, F. (2003). Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern recognition*, *36*(6), 1291–1302.

Cattaneo, M. E. G. V. (2003). Combining belief functions issued from dependent sources. In *proceedings of the 3rd International Symposium on Imprecise Probabilities and Their Applications (ISIPTA)* (Vol. 3, pp. 133–147).

Cho, S.-B., & Kim, J. H. (1995a). Combining multiple neural networks by fuzzy integral for robust classification. *Systems, Man and Cybernetics, IEEE Transactions on*, *25*(2), 380–384.

Cho, S.-B., & Kim, J. H. (1995b). Multiple network fusion using fuzzy logic. *IEEE Transactions on Neural Networks*, *6*(2), 497–501.

Crump, P. P. (1982). Statistical analysis: A computer oriented approach. *Technometrics*, *24*(3), 249–250.

Daum, F. (1996). Multitarget-multisensor tracking: principles and techniques. *Aerospace and Electronic Systems Magazine, IEEE*, *11*(2), 41–44.

Debie, E., Shafi, K., Lokan, C., & Merrick, K. (2013). Reduct based ensemble of learning classifier system for real-valued classification problems. In *proceedings of IEEE symposium on computational intelligence and ensemble learning* (pp. 66–73).

De Mántaras, R. L. (1991). A distance-based attribute selection measure for decision tree induction. *Machine learning*, *6*(1), 81–92.

Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *The annals of mathematical statistics*, 325–339.

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*(1), 79–111.

Denœux, T. (2006). The cautious rule of combination for belief functions and some extensions. In *proceedings of the 9th international conference on information fusion* (pp. 1–8).

Denœux, T. (2008). Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artificial Intelligence*, *172*(2), 234–264.

Denœux, T., & Zouhal, L. M. (2001). Handling possibilistic labels in pattern classification using evidential reasoning. *Fuzzy Sets and Systems*, *122*(3), 409-424.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *proceedings of the international workshop on multiple classifier systems* (pp. 1–15).

Dietterich, T. G., & Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research*, *2*, 263–286.

Dubes, R., & Jain, A. K. (1980). Clustering methodologies in exploratory data analysis. In *Advances in computers* (Vol. 19, pp. 113–228). Elsevier.

Dubois, D., & Prade, H. (1986). On the unicity of Dempster rule of combination. *International Journal of Intelligent Systems*, *1*(2), 133–142.

El-Monsef, M. A., Seddeek, M., & Medhat, T. (2003). Classification of sand samples according to radioactivity content by the use of euclidean and rough sets techniques. In *proceedings of the 4th nuclear and particle physics.*

Elouedi, Z., Mellouli, K., & Smets, P. (2001). Belief decision trees: theoretical foundations. *International Journal of Approximate Reasoning*, *28*(2), 91-124.

Fix, E., & Hodges Jr, J. L. (1951). *Discriminatory analysis-nonparametric discrimination: consistency properties* (Tech. Rep.). DTIC Document.

Franke, J., & Mandler, E. (1992). A comparison of two approaches for combining the votes of cooperating classifiers. In *proceedings of the 11th international conference on pattern recognition. vol. ii. conference b: Pattern recognition methodology and systems* (pp. 611–614).

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and computation*, *121*(2), 256–285.

Giacinto, G., & Roli, F. (2001). Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, *19*(9), 699–707.

Giacinto, G., Roli, F., & Fumera, G. (2000). Design of effective multiple classifier systems by clustering of classifiers. In *proceedings of the 15th international conference on pattern recognition* (Vol. 2, pp. 160–163).

Günter, S., & Bunke, H. (2004). Feature selection algorithms for the generation of multiple classifier systems and their application to handwritten word recognition. *Pattern recognition letters*, *25*(11), 1323–1336.

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence*(10), 993–1001.

Ho, T. K. (1995). Random decision forests. In *proceedings of the 3rd international conference on document analysis and recognition* (Vol. 1, pp. 278–282).

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, *20*(8), 832–844.

Hu, Q., Yu, D., Xie, Z., & Li, X. (2007). Eros: Ensemble rough subspaces. *Pattern recognition*, *40*(12), 3728–3739.

Huang, Y. S., & Suen, C. Y. (1993). The behavior-knowledge space method for combination of multiple classifiers. In *proceedings of the IEEE computer society conference on computer vision and pattern recognition* (pp. 347–347).

Hüllermeier, E. (2002). Possibilistic induction in decision-tree learning. In *proceedings of the machine learning* (pp. 173–184). Springer.

Jenhani, I., BenAmor, N., & Elouedi, Z. (2008). Decision trees as possibilistic classifiers. *International Journal of Approximate Reasoning*, *48*(3), 784–807.

Jenhani, I., Elouedi, Z., BenAmor, N., & Mellouli, K. (2005). Qualitative inference in possibilistic option decision trees. In *Symbolic and quantitative approaches to reasoning with uncertainty* (pp. 944–955). Springer.

Jiao, L., Denœux, T., & Pan, Q. (2015). Evidential editing k-nearest neighbor classifier. In *proceedings of the european conference on symbolic and quantitative approaches to reasoning and uncertainty* (pp. 461–471).

Johnson, D. S. (1973). Approximation algorithms for combinatorial problems. In *proceedings of the 5th annual ACM symposium on theory of computing* (pp. 38–49).

Jousselme, A., Grenier, D., & Bossé, E. (2001). A new distance between two bodies of evidence. *Information fusion*, *2*(2), 91-101.

Jousselme, A.-L., & Maupin, P. (2012). An evidential pattern matching approach for vehicle identification. In *Belief functions: Theory and applications* (pp. 45–52). Springer.

Jousselme, A.-L., & Maupin, P. (2013). Comparison of uncertainty representations for missing data in information retrieval. In *proceedings of the 16th international conference on information fusion* (pp. 1902–1909).

Kim, H., & Swain, P. H. (1995). Evidential reasoning approach to multisource-data classification in remote sensing. *IEEE Transactions on Systems, Man and Cybernetics*, *25*(8), 1257-1265.

Kim, Y. (2006). Toward a successful crm: variable selection, sampling, and ensemble. *Decision Support Systems*, *41*(2), 542–553.

Klein, J., Lecomte, C., & Miche, P. (2008). Preceding car tracking using belief functions and a particle filter. In *proceedings of the 19th international conference on pattern recognition* (pp. 1–4).

Komorowski, J., Øhrn, A., & Skowron, A. (2002). The rosetta rough set software system. *Handbook of data mining and knowledge discovery*, 2–3.

Kotsiantis, S., & Kanellopoulos, D. (2012). Combining bagging, boosting and random subspace ensembles for regression problems. *International Journal of Innovative Computing, Information and Control*, *8*(6), 3953–3961.

Kuncheva, L. (2000). *Fuzzy classifier design* (Vol. 49). Springer Science & Business Media.

Kuncheva, L. (2004). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

Kuncheva, L., Bezdek, J. C., & Duin, R. P. (2001). Decision templates for multiple classifier fusion: an experimental comparison. *Pattern recognition*, *34*(2), 299–314.

Kuncheva, L., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, *51*(2), 181–207.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436.

Lee, S. K. (1992). Imprecise and uncertain information in databases: An evidential approach. In *proceedings of the 8th international conference on data engineering* (pp. 614–621).

Lefèvre, E., Colot, O., & Vannoorenberghe, P. (2002). Belief function combination and conflict management. *Information fusion*, *3*(2), 149–162.

Lefèvre, E., & Elouedi, Z. (2013). How to preserve the conflict as an alarm in the combination of belief functions? *Decision Support Systems*, *56*, 326–333.

Lienemann, K., Plötz, T., & Fink, G. A. (2007). On the application of svm-ensembles based on adapted random subspace sampling for automatic classification of nmr data. In *proceedings of the international workshop on multiple classifier systems* (pp. 42–51).

Littlestone, N., & Warmuth, M. K. (1989). The weighted majority algorithm. In *proceedings of the 30th annual symposium on foundations of computer science* (pp. 256–261).

Mandler, E., & Shurmann, J. (1988). Combining the classification results of indipendent classifiers based on the Dempster Shafer theory of evidence. *Pattern recognition and artificial intellegence*, 381–393.

Martin, A. (2012). About conflict in the theory of belief functions. In *Belief functions: Theory and applications* (pp. 161–168). Springer.

Martínez-Muñoz, G., & Suárez, A. (2005). Switching class labels to generate classification ensembles. *Pattern Recognition*, *38*(10), 1483–1494.

Masson, M.-H., & Denœux, T. (2008). ECM: An evidential version of the fuzzy c-means algorithm. *Pattern Recognition*, *41*(4), 1384–1397.

Murphy, P., & Aha, D. (1996). UCI repository databases. *http://www.ics.uci.edu/mlear*.

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In *proceedings of the 22nd international conference on machine learning* (pp. 625–632).

Nutter, J. T. (1986). Uncertainty and probability.

Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, *11*, 169–198.

Pawlak, Z. (1982). Rough sets. *International Journal of Computer & Information Sciences*, *11*(5), 341–356.

Pawlak, Z. (1998). Rough set theory and its applications to data analysis. *Cybernetics & Systems*, *29*(7), 661–688.

Qin, B., Xia, Y., & Li, F. (2009). DTU: a decision tree for uncertain data. In *Advances in knowledge discovery and data mining* (pp. 4–15). Springer.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.

Quinlan, J. R. (1987). Decision trees as probabilistic classifiers. In *proceedings of the 4th international machine learning* (pp. 31–37).

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

Quost, B., Masson, M.-H., & Denœux, T. (2011). Classifier fusion in the dempster–shafer framework using optimized t-norm based combination rules. *International Journal of Approximate Reasoning*, *52*(3), 353–374.

Rhéaume, F., & Jousselme, A.-L. (2003). *Fusion of supervised classifiers using theory of evidence*.

Ristic, B., & Smets, P. (2006). The TBM global distance measure for the association of uncertain combat id declarations. *Information Fusion*, *7*(3), 276–284.

Rogova, G. (1994). Combining the results of several neural network classifiers. *Neural networks*, *7*(5), 777–781.

Saha, S., Murthy, C., & Pal, S. K. (2008). Classification of web services using tensor space model and rough ensemble classifier. In *proceedings of the international symposium on methodologies for intelligent systems* (pp. 508–513).

Samet, A., Lefèvre, E., & Ben Yahia, S. (2016). Evidential data mining: precise support and confidence. *Journal of Intelligent Information Systems*, *47*(1), 135-163.

Samet, A., Lefèvre, É., & Yahia, S. B. (2014). Evidential database: a new generalization of databases? In *proceedings of international conference on belief functions* (pp. 105–114).

Shafer, G. (1976). *A mathematical theory of evidence* (Vol. 42). Princeton university press.

Shi, L., Ma, X., Xi, L., Duan, Q., & Zhao, J. (2011). Rough set and ensemble learning based semi-supervised algorithm for text classification. *Expert Systems with Applications*, *38*(5), 6300–6306.

Shi, L., Xi, L., Ma, X., Weng, M., & Hu, X. (2011). A novel ensemble algorithm for biomedical classification based on ant colony optimization. *Applied Soft Computing*, *11*(8), 5674–5683.

Skalak, D. B. (1996). The sources of increased accuracy for two proposed boosting algorithms. In *proceedings of american association for artificial intelligence* (Vol. 1129, pp. 11–33).

Skowron, A., & Rauszer, C. (1992). The discernibility matrices and functions in information systems. In *Intelligent decision support* (pp. 331–362). Springer.

Skurichina, M., & Duin, R. P. (2002). Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis and Applications*, *5*(2), 121–135.

Smets, P. (1981). Medical diagnosis: Fuzzy sets and degrees of belief. *Fuzzy Sets and systems*, *5*(3), 259-266.

Smets, P. (1988). The Transferable Belief Model for quantified belief representation. *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, *1*, 267–301.

Smets, P. (1990). The combination of evidence in the Transferable Belief Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(5), 447-458.

Smets, P. (1995). The canonical decomposition of a weighted belief. In *proceedings of the international joint conference on artificial intelligence* (Vol. 14, pp. 1896–1901).

Smets, P. (1998). The application of the Transferable Belief Model to diagnostic problems. *International Journal of Intelligent Systems*, *13*, 127–157.

Srivastava, R., & Mock, T. (2002). *Belief functions in business decisions*. Physica-Verlag, Heidelberg, Springer-Verlag Company.

Straszecka, E. (2006). Combining uncertainty and imprecision in models of medical diagnosis. *Information Sciences*, *176*(20), 3026–3059.

Tessem, B. (1993). Approximations for efficient computation in the theory of evidence. *Artificial Intelligence*, *61*(2), 315–329.

Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2015a). Belief function combination: comparative study in classifier fusion framework. In *proceedings of the 1st International Symposium on Advanced Intelligent Systems and Informatics (AISI)* (Vol. 407, pp. 425–435).

Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2015b). Classifier fusion within the belief function framework using dependent combination rules. In *proceedings of the 22nd international symposium on methodologies for intelligent systems (ismis)* (Vol. 9384, pp. 133–138).

Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2016a). Feature selection from partially uncertain data within the belief function framework. In *proceedings of 16th international conference, IPMU 2016, eindhoven, the netherlands, june 20-24, 2016* (pp. 643–655).

Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2016b). Handling uncertain attribute values in decision tree classifier using the belief function theory. In *proceedings of the 17th international conference on artificial intelligence: Methodology, systems, and applications* (pp. 26–35).

Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2016c). New decision tree classifier for dealing with partially uncertain data. In *proceedings of the 25ème rencontres francophones sur la logique floue et ses applications* (pp. 57–64).

Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2017). Ensemble enhanced evidential k-nn classifier through random subspaces. In *proceedings of the european conference on symbolic and quantitative approaches to reasoning and uncertainty* (pp. 212–221).

Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2017). A novel k-nn approach for data with uncertain attribute values. In *proceedings of the 30th international conference on industrial engineering and other applications of applied intelligent systems, IEA/AIE 2017, part I* (pp. 160–170).

Trabelsi, A., Elouedi, Z., & Lefèvre, E. (2018). Ensemble enhanced evidential k-nn classifier through rough set reducts. In *proceedings of the 17th international conference on information processing and management of uncertainty in knowledge-based systems. theory and foundations* (pp. 383–394).

TRABELSI, S., & ELOUEDI, Z. (2008). Learning decision rules from uncertain data using rough sets. In *Computational intelligence in decision and control* (pp. 109–114). World Scientific.

Trabelsi, S., Elouedi, Z., & Lingras, P. (2011). Classification systems based on rough sets under the belief function framework. *International Journal of Approximate Reasoning*, *52*(9), 1409-1432.

Trabelsi, S., Elouedi, Z., & Mellouli, K. (2007). Pruning belief decision tree methods in averaging and conjunctive approaches. *International Journal of Approximate Reasoning*, *46*(3), 568–595.

Tresp, V., & Taniguchi, M. (1995). Combining estimators using non-constant weighting functions. In *proceedings of advances in neural information processing systems* (pp. 419–426).

Tumer, K., & Ghosh, J. (1996). Classifier combining: analytical results and implications. In *proceedings of the national conference on artificial intelligence* (pp. 126–132).

Tumer, K., & Oza, N. C. (2003). Input decimated ensembles. *Pattern Analysis & Applications*, *6*(1), 65–77.

Tupin, F., Bloch, I., & Maître, H. (1999). A first step toward automatic interpretation of sar images using evidential fusion of several structure detectors. *IEEE Transactions on Geoscience and Remote Sensing*, *37*(3), 1327-1343.

Turner, K., & Oza, N. C. (1999). Decimated input ensembles for improved generalization. In *proceedings of international joint conference on neural network (ijcnn'99)* (Vol. 5, pp. 3069–3074).

Umano, M., Okamoto, H., Hatono, I., Tamura, H., Kawachi, F., Umedzu, S., & Kinoshita, J. (1994). Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems. In *proceedings of the 3rd conference on fuzzy systems* (pp. 2113–2118).

Vannoorenberghe, P. (2004). On aggregating belief decision trees. *Information fusion*, *5*(3), 179–188.

Vannoorenberghe, P., & Denœux, T. (2002). Handling uncertain labels in multiclass problems using belief decision trees. In *proceedings of IPMU* (Vol. 3, pp. 1919–1926).

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, *10*(5), 988–999.

Wang, S.-L., Li, X., Zhang, S., Gui, J., & Huang, D.-S. (2010). Classification by combining PNN classifier ensemble with neighborhood rough set based gene reduction. *Computers in Biology and Medicine*, *40*(2), 179–189.

Weiss, S. M., & Kapouleas, I. (1989). An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. In *proceedings of the 11th international joint conference on artificial intelligence* (pp. 781–787). Morgan Kaufmann.

Xu, L., Krzyżak, A., & Suen, C. Y. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *Systems, man and cybernetics, IEEE transactions on*, *22*(3), 418–435.

Xu, P., Davoine, F., & Denœux, T. (2014). Evidential logistic regression for binary svm classifier calibration. In *proceedings of international conference on belief functions* (pp. 49–57).

Yager, R. R. (1987). On the dempster-shafer framework and new combination rules. *Information sciences*, *41*(2), 93–137.

Yao, Y., & Zhao, Y. (2009). Discernibility matrix simplification for constructing attribute reducts. *Information sciences*, *179*(7), 867–882.

Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC.

Zouhal, L. M., & Denœux, T. (1998). An evidence-theoretic k-nn rule with parameter optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *28*(2), 263–271.

Résumé:

Dans cette thèse, nous nous intéressons au problème de construction d'ensemble de classifieurs pour le traitement de données incertaines, plus particulièrement les données modélisées avec la théorie des fonctions de croyance. Dans un premier lieu, nous introduisons de nouveaux algorithmes d'apprentissage dans le cadre évidentiel. Par la suite, nous abordons le processus de construction d'ensemble de classifieurs qui se fonde sur deux étapes importantes : la sélection des classifieurs individuels et la fusion des classifieurs. En ce qui concerne l'étape de sélection, la diversité entre les classifieurs individuels est l'un des critères importants qui influe sur la performance de l'ensemble et peut être assurée en entraînant les classifieurs avec des divers sous ensembles d'attributs. Ainsi, nous proposons une nouvelle approche permettant l'extraction de sous ensembles d'attributs à partir des données décrites par des attributs évidentiels. Nous nous reposons principalement sur la théorie des ensembles approximatifs (rough set theory en anglais) pour identifier les différents sous ensembles d'attributs minimaux (connus en anglais sous le nom de reducts) permettant la même discrimination que l'ensemble des attributs intial. Nous développons ensuite trois méthodes permettant la sélection des reducts les plus appropriés pour un système d'ensemble. Une évaluation de nos trois méthodes de sélection de reducts a été effectuée et la meilleure méthode a été utilisée pour la sélection des classifeurs individuels. En ce qui concerne la phase de fusion, nous proposons de sélectionner l'opérateur de fusion le plus approprié parmi les règles les plus connues à savoir la règle de combinaison de Dempster, la règle prudente et la règle t-norm optimisée.

Mots clés: Ensemble des classifieurs, théorie des fonctions de croyance, attributs evidentiels, théorie des ensembles approximatifs, sélection des classifieurs, fusion des classifieurs.

Abstract

The work presented in this Thesis concerns the construction of ensemble classifiers for addressing uncertain data, precisely data with evidential attributes. We start by developing newest machine learning classifiers within an evidential environment and then we tackle the ensemble construction process which follows two important steps: base individual classifier selection and classifier combination. Regarding the selection step, diversity between the base individual classifiers is one among the important criteria impacting the ensemble performance and it can be achieved by training the base classifiers on diverse feature subspaces. Thus, we propose a novel framework for feature subspace extraction form data with evidential attributes. We mainly relied on the rough set theory for identifying all possible minimal feature subspaces, called reducts, allowing the same discrimination as the whole feature set. Then, we develop three methods enabling the selection of the most suitable diverse reducts for an ensemble of evidential classifiers. The proposed reduct selection methods are evaluated according to several assessment criteria and the best one is used for selecting the best individual classifiers. Concerning the integration level, we propose to select the most appropriate combination operator among some well-known ones, including the Dempster, the cautious and the optimized t-norm based rules.

keywords Ensemble classifiers, belief function theory, evidential attributes, rough set reducts, classifier selection, classifier fusion.