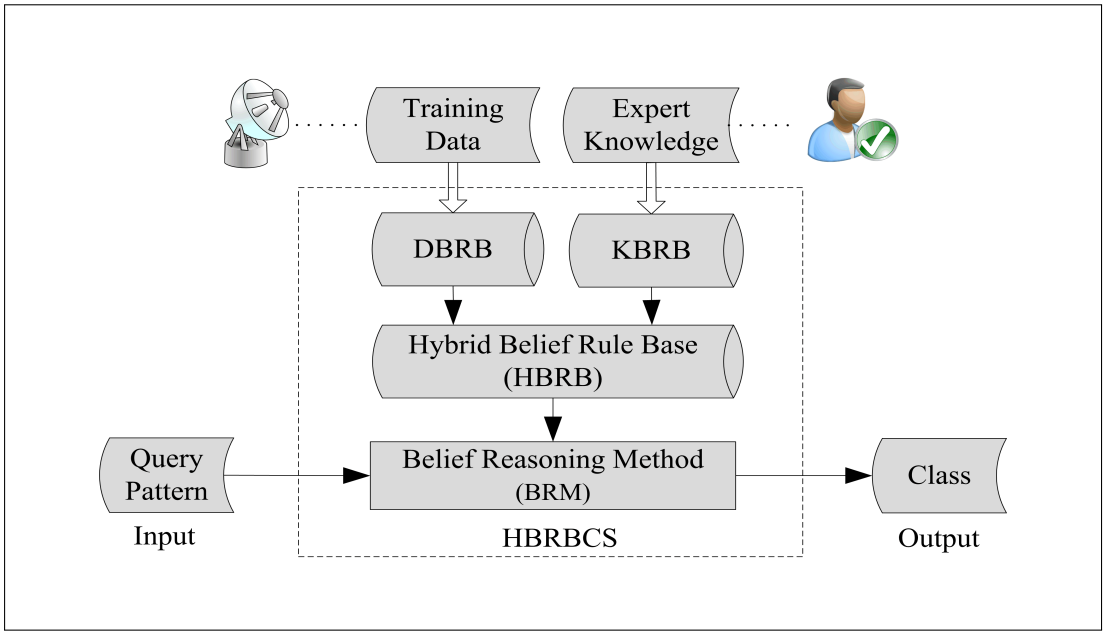
par **Lianmeng JIAO**

Classification of uncertain data in the framework of belief functions: Nearest-neighbor-based and rule-based approaches

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC.



Soutenue le : 26 Octobre 2015
Spécialité : Technologies de l'Information et des Systèmes

Classification of uncertain data in the framework of belief functions: Nearest-neighbor-based and rule-based approaches

Thèse soutenue le 26 Octobre 2015 devant le jury composé de :

MARTIN Arnaud	Professor	Université de Rennes 1	(Rapporteur)
JI Hongbing	Professor	Xidian University	(Rapporteur)
DAVOINE Franck	Researcher	Université de Technologie de Compiègne	(Examinateur)
CHENG Yongmei	Professor	Northwestern Polytechnical University	(Examinateur)
DENŒUX Thierry	Professor	Université de Technologie de Compiègne	(Directeur de thèse)
PAN Quan	Professor	Northwestern Polytechnical University	(Directeur de thèse)

To
my parents,
my beloved,
my brother and sister,
for your love, support and encouragement.

Abstract

In many classification problems, data are inherently uncertain. The available training data might be imprecise, incomplete, even unreliable. Besides, partial expert knowledge characterizing the classification problem may also be available. These different types of uncertainty bring great challenges to classifier design. The theory of belief functions provides a well-founded and elegant framework to represent and combine a large variety of uncertain information. In this thesis, we use this theory to address the uncertain data classification problems based on two popular approaches, i.e., the k-nearest neighbor rule (kNN) and rule-based classification systems.

For the kNN rule, one concern is that the imprecise training data in class overlapping regions may greatly affect its performance. An evidential editing version of the kNN rule was developed based on the theory of belief functions in order to well model the imprecise information for those samples in overlapping regions. Another consideration is that, sometimes, only an incomplete training data set is available, in which case the ideal behaviors of the kNN rule degrade dramatically. Motivated by this problem, we designed an evidential fusion scheme for combining a group of pairwise kNN classifiers developed based on locally learned pairwise distance metrics.

For rule-based classification systems, in order to improving their performance in complex applications, we extended the traditional fuzzy rule-based classification system in the framework of belief functions and develop a belief rule-based classification system to address uncertain information in complex classification problems. Further, considering that in some applications, apart from training data collected by sensors, partial expert knowledge can also be available, a hybrid belief rule-based classification system was developed to make use of these two types of information jointly for classification.

Keywords Data classification, Information fusion, Uncertainty management, Theory of belief functions, k-nearest neighbor rule, rule-based classification system

Résumé

Dans de nombreux problèmes de classification, les données sont intrinsèquement incertaines. Les données d'apprentissage disponibles peuvent être imprécises, incomplètes, ou même peu fiables. En outre, des connaissances spécialisées partielles qui caractérisent le problème de classification peuvent également être disponibles. Ces différents types d'incertitude posent de grands défis pour la conception de classifieurs. La théorie des fonctions de croyance fournit un cadre rigoureux et élégant pour la représentation et la combinaison d'une grande variété d'informations incertaines. Dans cette thèse, nous utilisons cette théorie pour résoudre les problèmes de classification des données incertaines sur la base de deux approches courantes, à savoir, la méthode des k plus proches voisins (kNN) et la méthode à base de règles.

Pour la méthode kNN, une préoccupation est que les données d'apprentissage imprécises dans les régions où les classes de chevauchent peuvent affecter ses performances de manière importante. Une méthode d'édition a été développée dans le cadre de la théorie des fonctions de croyance pour modéliser l'information imprécise apportée par les échantillons dans les régions qui se chevauchent. Une autre considération est que, parfois, seul un ensemble de données d'apprentissage incomplet est disponible, auquel cas les performances de la méthode kNN se dégradent considérablement. Motivé par ce problème, nous avons développé une méthode de fusion efficace pour combiner un ensemble de classifieurs kNN couplés utilisant des métriques couplées apprises localement.

Pour la méthode à base de règles, afin d'améliorer sa performance dans les applications complexes, nous étendons la méthode traditionnelle dans le cadre des fonctions de croyance. Nous développons un système de classification fondé sur des règles de croyance pour traiter des informations incertains dans les problèmes de classification complexes. En outre, dans certaines applications, en plus de données d'apprentissage, des connaissances expertes peuvent également être disponibles. Nous avons donc développé un système de classification hybride fondé sur des règles de croyance permettant d'utiliser ces deux types d'information pour la classification.

Mots-clés Classification, Fusion d'informations, Gestion de l'incertitude, Théorie des fonctions de croyance, k plus proches voisins, classification à base de règles

Acknowledgements

This thesis was carried out at the Heudiasyc laboratory of Université de Technologie de Compiègne (UTC) and the Information Fusion Technology laboratory of Northwestern Polytechnical University (NPU), under the support of the China Scholarship Council.

I would like to express my sincere gratitude to my supervisor at UTC, Prof. Thierry Denœux, for his help and encouragement throughout this Ph.D. research. As an expert in the fields of belief functions and machine learning, he provided me many useful ideas for my Ph.D. work. I always discussed with him about my research and I was inspired by his constructive comments. I have also learned a great deal from his instructions for developing a rigorous way of research.

I would also like to gratefully thank my supervisor at NPU, Prof. Quan Pan. I have been working with him since my Master program and he has led me into this interesting research field. He provided me with great support for my overseas study, as well as attending national and international conferences. When I encountered difficulties in work or in life, he always provided me great help and encouragement. His supervision is quit important and useful for me to finish this thesis successfully.

My thanks also go to the researchers at UTC and NPU, Sébastian Destercke, Yves Grandvalet, Philippe Xu, Yongmei Cheng, Yan Liang, Feng Yang and Zhunga Liu, with whom I had many insightful discussions about many aspects of research. I also wish to thank all the friends with whom I shared much wonderful time both in Xi'an and Compiène.

Finally, I wish to express my deepest appreciation to my parents, my girlfriend Xiaoxue, my brother and sister, for their love, support and encouragement throughout the long journey of my study.

1st September 2015
Lianmeng Jiao at Compiègne

Publications

International journals

[1] L. Jiao, Q. Pan, T. Denœux, Y. Liang and X. Feng. Belief rule-based classification system: Extension of FRBCS in belief functions framework. *Information Sciences*, vol.309, pp.26-49, 2015.

[2] L. Jiao, Q. Pan and X. Feng. Multi-hypothesis nearest-neighbor classifier based on class-conditional weighted distance metric. *Neurocomputing*, vol.151, pp.1468-1476, 2015.

[3] L. Jiao, Q. Pan, Y. Liang, X. Feng and F. Yang. Combining sources of evidence with reliability and importance for decision making. *Central European Journal of Operations Research*, DOI: 10.1007/s10100-013-0334-3, 2013.

International conferences

[1] L. Jiao, T. Denœux and Q. Pan. Evidential editing k-nearest neighbor classifier. In *Proceedings of the 13th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 461-471, Compiègne, France, 2015.

[2] L. Jiao, T. Denœux and Q. Pan. Fusion of pairwise nearest-neighbor classifiers based on pairwise-weighted distance metric and Dempster-Shafer theory. In *Proceedings of the 17th International Conference on Information Fusion*, pages 1-7, Salamanca, Spain, 2014.

[3] L. Jiao, Q. Pan, X. Feng and F. Yang. An evidential k-nearest neighbor classification method with weighted attributes. In *Proceedings of the 16th International Conference on Information Fusion*, pages 145-150, Istanbul, Turkey, 2013.

[4] L. Jiao, Q. Pan, Y. Liang and F. Yang. A nonlinear tracking algorithm with range-rate measurements based on unbiased measurement conversion. In *Proceedings of the 15th International Conference on Information Fusion*, pages 1400-1405, Singapore, 2012.

Contents

List of Tables

List of Figures

Introduction

Data classification, also called supervised learning, is the process of predicting the class label of a new instance based on a set of labeled samples. It occurs in a wide range of human activities. In many real-world applications, data contains inherent uncertainty [21,100]. For example, the data may be incomplete, which refers to cases where the value of a variable is missing. Sometimes, the data may be imprecise, when the value of a variable is given, but not with enough precision. In addition, the data may be unreliable, i.e., the obtained values might be wrong. The above types of uncertainty can be caused by a variety of factors, such as the random nature of the physical data generation and collection processes, measurement and decision errors, insufficient knowledge, etc. [74]. These widely existed uncertainties present great challenges to classifier design.

As a research field closely related to real-world applications, in the past several decades, a wide variety of approaches have been proposed to cope with data classification problem. Among them, the k-nearest neighbor rule, first developed by Fix and Hodges [38], has become one of the most popular statistical classification techniques. As a non-parametric lazy learning algorithm, its classification process is quite simple and it does not depend on the underlying data distribution. Apart from the traditional statistical approaches, some computational intelligence-based classification approaches have also been developed to mimic the human reasoning process. One of the most representative approaches is rule-based classification [17], which classifies a new instance based on a set of rules learnt from training samples or expert knowledge. The most useful characteristic of this approach is high interpretability due to the used linguistic model. In this thesis, our work mainly focuses on these two approaches, i.e., the k-nearest neighbor rule and rule-based classification system.

For the k-nearest neighbor (kNN) rule, it has been proved that its error rate approaches the optimal Bayes error rate asymptotically. However, in the finite sample case, its performance may greatly affected by the imperfect training data. One consideration is that the patterns from different classes overlap largely. Though the training samples in overlapping regions are assigned with precise labels, they actually cannot be seen as truly representatives of their corresponding clusters. Therefore, there is a need for well modeling the imprecise information for those samples in overlapping regions. Another consideration is that sometimes only an incomplete training data set is available. In these situations,

the ideal performances of the kNN rule degrade dramatically. Therefore, obtaining good performances based on incomplete training data set is also a critical issue for classifier design.

For rule-based classification systems, the used linguistic model makes this approach interpretable to users. However, every coin has two sides. The rule-based classification systems may face lack of accuracy in some complex applications, due to the lack of flexibility of the linguistic model. Therefore, there is a real need for improving the performance of the rule-based classification systems in complex classification problems. In additions, in some real-world classification problems, apart from training data collected by sensors, partial expert knowledge provided by humans may also be available. These two types of information are usually independent but complementary. Thus, we need a hybrid classification model that can make use of these two types of uncertain information jointly.

The traditional classification approaches usually work within the probabilistic framework. However, probability theory only captures the randomness aspect of the data, but neither imprecision, nor incompleteness which are inherent in uncertain data. Therefore, many theories have been developed during the last decades to construct more powerful representations. In this thesis, we focus on the theory of belief functions, also known as Dempster-Shafer theory [24, 96], to address the uncertain data classification problems. As a generalization of probability theory, it offers a well-founded and workable framework to represent and combine a large variety of uncertain information. We consider data classification from an information fusion point of view by combining the evidence constructed from uncertain training data or expert knowledge.

This thesis is structured in three parts:

Part I introduces the theoretical background that supports the thesis and provides a literature review for related work. Chapter 1 recalls fundamental aspects of belief functions. We describe how information is represented and combined in the framework of belief functions. Some useful operations as well as decision rules are also presented. Chapter 2 presents general aspects of classification for uncertain data. After a discussion about the inherent properties of uncertain data, we formulate four critical issues concerning the uncertainty in data classification field, which are the main concerns of this thesis. Then, we provide a literature review for related work, emphasizing on nearest-neighbor-based and rule-based approaches.

Part II focuses on classification of uncertain data using nearest-neighbor-based approaches. Chapter 3 is oriented to performance enhancement of the kNN rule for cases in which patterns from different classes overlap strongly. In contrast, in Chapter 4, we are mainly concerned the classification problems based on incomplete training data sets, which is a key issue for nearest-neighbor-based approaches.

Part III focuses on classification of uncertain data using rule-based approaches. In Chapter 5, we present a method for improving the performance of the rule-based classification system in harsh working conditions, where only partially reliable training data are available. Finally, in Chapter 6, we propose a method for handling expert knowledge together with training data in rule-based systems.

Part I

Theoretical background and literature review

The purpose of this fist part is to introduce the theoretical background and to provide a literature review for related work.

Chapter 1 is intended to provide a detailed introduction about the theory of belief functions, including some basic concepts and some useful operations.

Chapter 2 is designed to formulate the uncertain data classification problems considered in this thesis, and to introduce some existed work for classification of uncertain data using nearest-neighbor-based and rule-based approaches.

Theory of belief functions

Uncertain data classification tasks require powerful tools to represent and combine different types of uncertain information. The theory of belief functions [24, 96], also known as Dempster-Shafer theory or evidence theory, is a generalization of probability theory. It offers a well-founded and workable framework to represent and combine a large variety of uncertain information [125]. It is also a generalization of possibility theory [133] and is closely linked to other theories including fuzzy sets [132], random sets [77] and imprecise probability [117].

In this chapter, we first describe in Section 1.1 how different types of information can be represented in the framework of belief functions. In Section 1.2, we discuss the combination of mass functions and present some widely-used combination rules. Next, we describe in Section 1.3 some operations over the frame of discernment, such as conditioning, deconditioning and discounting. Then, in Section 1.4, we consider the issue of decision making using belief functions. Finally, Section 1.5 concludes this chapter.

1.1 Representation of evidence

The prerequisite of reasoning in the framework of belief functions is the representation of the available information, which is usually called *evidence*. This is done based on some basic functions used to represent our knowledge about the considered problem. At a glance, there are three main functions: mass, belief and plausibility functions. The mass function is the most basic and intuitive way of expressing someone's degrees of belief. The belief and plausibility functions are often used to compute intervals in order to bound the uncertainty. We will see their usefulness and their expressive power in the following sections.

1.1.1 Mass function

In the theory of belief functions, a problem domain is represented by a finite set $\Omega = \{\omega_1, \omega_2, \cdots, \omega_n\}$ of mutually exclusive and exhaustive hypotheses called the *frame of discernment*. Given a piece of evidence held by an agent, the state of belief about the

truth of the problem is represented by a *mass function* or *basic belief assignment* (BBA) defined as follows [96].

Definition 1.1 (mass function)**.** *A mass function, over a frame of discernment* Ω*, is a mapping function* m*:* $2^\Omega \to [0,1]$*, such that*

$$m(\emptyset) = 0 \quad \text{and} \quad \sum_{A \subseteq \Omega} m(A) = 1. \tag{1.1}$$

Elements $A \subseteq \Omega$ having $m(A) > 0$ are called the *focal sets* of the mass function m. Each number $m(A)$ represents the part of belief assigned to the hypothesis that the truth ω lies in the subset A (i.e., the hypothesis $\omega \in A$). It is important to understand that the hypothesis $\omega \in A$ does not support the membership of ω to any subset $B \subsetneq A$. The belief assigned to Ω, or $m(\Omega)$, is referred to as the degree of *ignorance*. Definition 1.1 imposes that the empty set cannot be a focal set. For the open-world assumption [103], the quantity $m(\emptyset)$ is interpreted as the degree of support of the hypothesis that the truth ω is actually outside of the frame Ω. In this thesis, the closed-world assumption is used, i.e., the frame of discernment is considered exhaustive.

To interpret how the formalism of mass function can be used to represent a piece of evidence, we provide the following murder example [30] for illustration.

Exemple 1.1 (murder)**.** *A murder has been committed and there are three suspects: Peter, John and Mary. The question* Q *of interest is the identity of the murderer. In the framework of belief functions, the set* $\Omega = \{Peter, John, Mary\}$ *can be seen as the frame of discernment for the considered problem. The piece of evidence under study is a testimony: a witness saw the murderer and because he is short-sighted, he can only report that he saw a man. However, this testimony is not fully reliable because we know that the witness is drunk 20% of the time. How can such a piece of evidence be encoded in the language of mass functions?*

We can see that what the testimony tells us about Q *depends on the answer to another question* Q'*: Was the witness drunk at the time of the murder? If he was not drunk, we know that the murderer is Peter or John. Otherwise, we know nothing. Since there is 80% chance that the former hypothesis holds, we assign a 0.8 mass to the set* $\{Peter, John\}$*, and 0.2 to the frame of discernment* Ω*, yielding the following mass function:*

$$m(\{Peter, John\}) = 0.8, \quad m(\Omega) = 0.2.$$

The mass function in Definition 1.1 has several special cases, which are quite useful to encode different types of information. A mass function is said to be

- *categorical*, if it has only one focal set A (denoted as m_A). This can be interpreted as being certain that the truth lies in A.

- *Bayesian*, if all of its focal sets are singletons (i.e., sets with cardinality equal to one). In this case, the mass function reduces to the precise probability distribution.
- *dogmatic*, if Ω is not a focal set. In this case, there is no ignorance about the state of belief for the truth of the problem.
- *vacuous*, if Ω is the only focal set (denoted as m_Ω). This situation corresponds to complete ignorance as the closed-world assumption implies that hypothesis $\omega \in \Omega$ is always true.
- *simple*, if it has at most two focal sets and one of them is Ω if it has two. It is denoted by A^w, A being the focal set different from Ω and $1-w$ the confidence that the truth lies in A. The vacuous mass function can thus be noted as A^1 for any $A \subset \Omega$, and a categorical mass function can be noted as A^0 for some $A \neq \Omega$.

In Table 1.1, we show some examples of these special mass functions and the types of information that are encoded.

Table 1.1: Examples of mass functions defined over $\Omega = \{\omega_1, \omega_2, \omega_3\}$

Mass function	Example	Type of information
Categorical	$m(\{\omega_1, \omega_2\}) = 1$	Certain information
Bayesian	$m(\{\omega_1\}) = m(\{\omega_2\}) = m(\{\omega_3\}) = 1/3$	Precise information
Dogmatic	$m(\{\omega_1\}) = 1/2$, $m(\{\omega_2, \omega_3\}) = 1/2$	Information without ignorance
Vacuous	$m(\Omega) = 1$	Complete ignorance
Simple	$m(\{\omega_1\}) = 1/2$, $m(\Omega) = 1/2$	Unique hypothesis with partial confidence

1.1.2 Belief and plausibility functions

In addition to mass function, there are two other important functions to represent evidence: the belief function Bel and the plausibility function Pl [96].

Definition 1.2 (belief and plausibility functions)**.** *The belief and plausibility functions, over a frame of discernment Ω, are defined, respectively, as*

$$Bel(A) = \sum_{B \subseteq A} m(B), \quad \forall A \subseteq \Omega. \tag{1.2}$$

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \quad \forall A \subseteq \Omega. \tag{1.3}$$

The degree of belief $Bel(A)$ quantifies the amount of justified specific support to be given to A. It can be interpreted as the degree of belief that the truth lies in A. Note

that this definition is close to that of a mass function. The difference lies in the fact that the degree of belief $Bel(A)$ can be further divided between some subsets of A. The degree of plausibility $Pl(A)$ quantifies the maximum amount of potential specific support that could be given to A. It can be interpreted as the belief that fails to doubt A. The interval $[Bel(A), Pl(A)]$ can be seen as the lower and upper bounds of support to A.

Properties

1. $Bel(A) \leq Pl(A), \quad \forall A \subseteq \Omega$;

2. Bel and Pl are related by:

$$Bel(A) = 1 - Pl(\overline{A}), \quad \forall A \subseteq \Omega, \tag{1.4}$$

$$Pl(A) = 1 - Bel(\overline{A}), \quad \forall A \subseteq \Omega; \tag{1.5}$$

3. A mass function m can be expressed in terms of Bel and Pl in the following way:

$$m(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|} Bel(B), \quad \forall A \subseteq \Omega, \tag{1.6}$$

$$m(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|+1} Pl(\overline{B}), \quad \forall A \subseteq \Omega. \tag{1.7}$$

From the definitions and properties of the belief and plausibility functions, we can see that there exists a one-to-one correspondence between mass, belief and plausibility functions as shown in Figure 1.1. To provide a graphical explanation of belief and plausibility functions and their relation to the mass function, we consider the following example.

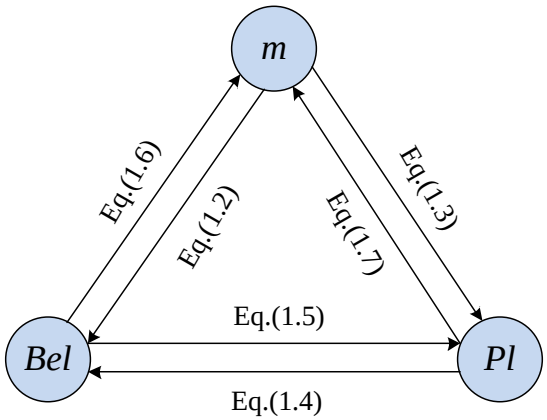


Figure 1.1: One-to-one correspondence between mass (m), belief (Bel) and plausibility (Pl) functions

Exemple 1.2. *As shown in Figure 1.2, we consider a mass function m with five focal sets B_i, $i = 1, 2, 3, 4, 5$. Then, the degree of belief in A is*

$$Bel(A) = m(B_1) + m(B_2),$$

whereas the plausibility of A is

$$Pl(A) = m(B_1) + m(B_2) + m(B_3) + m(B_4).$$

The degree of belief in the complement of A is

$$Bel(\overline{A}) = m(B_5),$$

which is clearly equal to $1 - Pl(A)$. The plausibility in the complement of A is

$$Pl(\overline{A}) = m(B_3) + m(B_4) + m(B_5),$$

which is clearly equal to $1 - Bel(A)$.

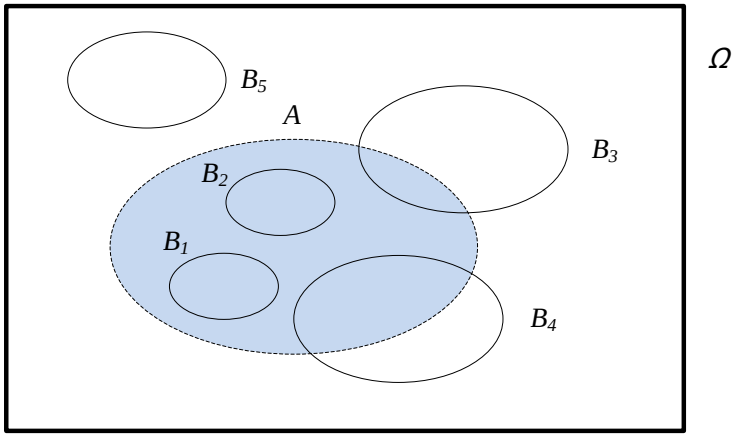


Figure 1.2: Graphical example of mass, belief and plausibility functions

1.1.3 Distance between mass functions

With two sources of evidence S_1 and S_2 characterizing the same considered problem, sometimes we need to know the dissimilarity between them. It can be characterized by the distance between their corresponding mass functions. Since the introduction of Dempster's conflict measure [24], many distance measures between mass functions have been defined in the literature [54]. Here, we present the definition of Jousselme's distance d_J [53], which is one of the most commonly used distances.

Definition 1.3 (Jousselme's distance)**.** *Let m_1 and m_2 be two mass functions defined over the same frame of discernment Ω, containing n mutually exclusive and exhaustive hypotheses. The Jousselme's distance between m_1 and m_2 is*

$$d_J(m_1, m_2) = \sqrt{\frac{1}{2}(\overrightarrow{m}_1 - \overrightarrow{m}_2)^T \mathbf{D}(\overrightarrow{m}_1 - \overrightarrow{m}_2)}, \tag{1.8}$$

where $\overrightarrow{m}_1$ *and* $\overrightarrow{m}_2$ *are column vectors composed of the mass for all of the* 2^n *subsets of* Ω*, and* $\mathbf{D}$ *is a* $2^n \times 2^n$ *matrix with elements given by* $D_{i,j} = \dfrac{|A_i \cap B_j|}{|A_i \cup B_j|}$*,* $A_i, B_j \in 2^\Omega$*. The factor* $1/2$ *is used to normalize* d_J *so that* $0 \leq d_J(m_1, m_2) \leq 1$*.*

From the above definition, another way to write d_J is:

$$d_J(m_1, m_2) = \sqrt{\frac{1}{2}(\|\overrightarrow{m}_1\|^2 + \|\overrightarrow{m}_2\|^2) - \langle \overrightarrow{m}_1, \overrightarrow{m}_2 \rangle}, \tag{1.9}$$

where $\|\overrightarrow{m}\|^2$ is the square norm of $\overrightarrow{m}$, and $\langle \overrightarrow{m}_1, \overrightarrow{m}_2 \rangle$ is the scalar product defined by

$$\langle \overrightarrow{m}_1, \overrightarrow{m}_2 \rangle = \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} m_1(A_i) m_2(B_j) \frac{|A_i \cap B_j|}{|A_i \cup B_j|},$$

with $A_i, B_j \in 2^\Omega$ for $i, j = 1, 2, \cdots, 2^n$.

1.2 Combination of evidence

In the preceding section, we have presented some functions that allow us to represent the available pieces of evidence. The next step of the reasoning process is to combine these information to obtain a single mass function. In this section, we will introduce some popular combination rules that allow us to aggregate knowledge held by several pieces of evidence into a single one. The differences between these combination rules mainly depend on two issues: the reliability of the sources of information and the degree of dependence between them.

1.2.1 Dempster's rule

Given two mass functions m_1 and m_2 induced from two sources of information, a combination rule yields a new mass function that represents our new state of knowledge after taking into consideration both sources of information. Dempster's rule is the most popular alternative to combine several distinct bodies of evidence [24].

Definition 1.4 (Dempster's rule)**.** *Let* m_1 *and* m_2 *be two mass functions defined over the same frame of discernment* Ω*. One can combine them using Dempster's rule to compute a new mass function defined by*

$$(m_1 \oplus m_2)(A) = \begin{cases} 0, & \text{for } A = \emptyset \\ \dfrac{1}{1-\kappa} \displaystyle\sum_{B \cap C = A} m_1(B) m_2(C), & \text{for } A \subseteq \Omega, A \neq \emptyset, \end{cases} \tag{1.10}$$

with

$$\kappa = \sum_{B \cap C = \emptyset} m_1(B) m_2(C). \tag{1.11}$$

The quantity κ measures the conflict between the two mass functions m_1 and m_2. The combination rule is valid only if $\kappa < 1$, otherwise, m_1 and m_2 are inconsistent and cannot be combined.

Properties

1. Commutativity: $m_1 \oplus m_2 = m_2 \oplus m_1$;
2. Associativity: $(m_1 \oplus m_2) \oplus m_3 = m_1 \oplus (m_2 \oplus m_3)$;
3. It has the vacuous mass function m_Ω as the unique neutral element: $m \oplus m_\Omega = m_\Omega \oplus m = m$;

Next, let us come back to the murder example studied in Example 1.1 and illustrate how Dempster's rule works to combine distinct bodies of evidence.

Exemple 1.3 (murder continued)**.** *Remember that the first item of evidence gave us the following mass function:*

$$m_1(\{Peter, John\}) = 0.8, \quad m_1(\Omega) = 0.2,$$

concerning the murderer over the frame $\Omega = \{Peter, John, Mary\}$*. Let us now assume that we have a new piece of evidence: a blond hair has been found. This new evidence supports the hypothesis that the murderer is either John or Mary, as they are blond while Peter is not. However, this piece of evidence is reliable only if the room has been cleaned before the crime. If we judge that there is* 60% *chance that it was the case, then our second piece of evidence is modeled by the following mass function:*

$$m_2(\{John, Mary\}) = 0.6, \quad m_2(\Omega) = 0.4.$$

With the above two distinct pieces of evidence, Dempster's rule can be used to combine them into an integrated mass function as

$$m(\{John\}) = 0.48, \;\; m(\{Peter, John\}) = 0.32, \;\; m(\{John, Mary\}) = 0.12, \;\; m(\Omega) = 0.08.$$

From the above combination results, we can see that each focal set of m *is obtained by intersecting one focal set of* m_1 *and one focal set of* m_2*. Consequently, the integrated mass function* m *is more focussed than both* m_1 *and* m_2*. Therefore, after combination, we can obtain a more precise answer about the considered problem.*

1.2.2 Cautious rule

A strong assumption of Dempster's rule is that the pieces of evidence to be combined are independent. However, this is not always verified in practice. In cases where this

independence assumption is not reasonable, Denœux [29] proposed to use the cautious rule. Before defining the cautious rule, let us first show the canonical decomposition of a mass function.

Shafer [96] defined a *separable mass function* as the result of the $\oplus$ combination of simple mass functions. If this separable mass function m is non-dogmatic, it can be uniquely represented by

$$m = \bigoplus_{\emptyset \neq A \subset \Omega} A^{w(A)}, \tag{1.12}$$

with $w(A) \in [0, 1]$ for all $A \subset \Omega$, $A \neq \emptyset$. This representation is called the *canonical decomposition* of m. The weights $w(A)$ can be obtained from the commonalities as follows:

$$w(A) = \prod_{B \supseteq A} q(B)^{(-1)^{|B|-|A|+1}}, \quad \forall A \subset \Omega, \tag{1.13}$$

where the commonalities $q(B) = \sum_{C \supseteq B} m(C)$, $\forall B \subseteq \Omega$. The above general formula for calculating weights $w(A)$ seems complicated. In [29], Denœux provided a simple analytical formula for the weight function associated to the following special mass function. Let $A_1, A_2, \cdots, A_n$ be n subsets of Ω such that $A_i \cap A_j = \emptyset$, for all $i, j \in \{1, 2, \cdots, n\}$, and let m be a mass function on Ω with focal sets $A_1, A_2, \cdots, A_n$, and Ω. The weight function associated to m is

$$w(A) = \begin{cases} \dfrac{m(\Omega)}{m(A_k) + m(\Omega)}, & \text{if } A = A_k \\ 1, & \text{otherwise.} \end{cases} \tag{1.14}$$

Definition 1.5 (cautious rule)**.** *Let m_1 and m_2 be two non-dogmatic mass functions defined over the same frame of discernment Ω, and w_1 and w_2 the associated weight functions from their respective canonical decompositions. Their combination using the cautious rule is defined as*

$$m_1 \mathbin{⍀} m_2 = \bigoplus_{\emptyset \neq A \subset \Omega} A^{w_1(A) \wedge w_2(A)}, \tag{1.15}$$

where $\wedge$ denotes the minimum operator.

Compared to Dempster's rule, besides the commutativity and associativity, the cautious rule is also *idempotent* (i.e., $m \mathbin{⍀} m = m$), which is a natural requirement for a rule allowing the combination of dependent pieces of evidence.

1.2.3 Triangular norm-based rules

It is possible to formulate both Dempster's rule and the cautious rule with a triangular norm-based combination rule [29]. The combination of two non-dogmatic mass functions m_1 and m_2 with Dempster's rule can be written as an expression similar to Eq. (1.15):

$$m_1 \oplus m_2 = \bigoplus_{\emptyset \neq A \subset \Omega} A^{w_1(A) w_2(A)}. \tag{1.16}$$

With Dempster's rule, the weights $w_1(A)$ and $w_2(A)$ are multiplied, whereas with the cautious rule, the minimum operator is used. Frank's parameterized family of t-norms [56] generalizing these two operators is defined as

$$a \top_s b = \begin{cases} a \wedge b, & \text{if } s = 0 \\ ab, & \text{if } s = 1 \\ \log_s \left(1 + \dfrac{(s^a - 1)(s^b - 1)}{s - 1}\right), & \text{otherwise,} \end{cases} \tag{1.17}$$

for all $a, b \in [0, 1]$, where s is a positive parameter. For any $s \in (0, 1)$, $a \top_s b$ returns a value between ab and $a \wedge b$.

Definition 1.6 (triangular norm-based rules)**.** *Let m_1 and m_2 be two non-dogmatic mass functions defined over the same frame of discernment Ω, and w_1 and w_2 the associated weight functions from their respective canonical decompositions. Their combination using the Frank's family of triangular norm-based rules is defined as*

$$m_1 \circledast_s m_2 = \bigoplus_{\emptyset \neq A \subset \Omega} A^{w_1(A) \top_s w_2(A)}, \tag{1.18}$$

where $\top_s$ is Frank's parameterized family of t-norms with $s \in [0, 1]$.

Obviously, when $s = 0$, the triangular norm-based rule corresponds to the cautious rule (i.e., $\circledast_0 = \owedge$), and when $s = 1$, it corresponds to Dempster's rule (i.e., $\circledast_1 = \oplus$). All these rules inherit important properties from t-norms: they are commutative and associative, and they admit the vacuous mass function as neutral element.

1.2.4 Other alternative combination rules

For the above reviewed combination rules, the sources of evidence to be combined are assumed to be reliable. We could, however, make different assumptions about the reliability of the two sources. For instance, we could assume that at least one of them is reliable. This assumption results in the following binary operation, called the *disjunctive rule of combination* [101]:

$$(m_1 \cup m_2)(A) = \sum_{B \cup C = A} m_1(B) m_2(C), \text{ for } A \subseteq \Omega. \tag{1.19}$$

This operation is clearly commutative and associative, and it does not have a neutral element. Combining mass functions disjunctively can be seen as a conservative strategy, as the disjunctive rule relies on a weaker assumption about the reliability of the sources, as compared to Dempster's rule. However, mass functions become less and less focussed as more pieces of evidence are combined using the disjunctive rule. In general, the disjunctive rule may be preferred in case of heavy conflict between the different pieces of evidence.

An alternative rule, which is somehow intermediate between the disjunctive rule and Dempster's rule, has been proposed by Dubois and Prade [35]. It is defined as:

$$(m_1 \star m_2)(A) = \begin{cases} 0, & \text{for } A = \emptyset \\ \sum\limits_{B \cap C = A} m_1(B) m_2(C) + \sum\limits_{B \cap C = \emptyset, B \cup C = A} m_1(B) m_2(C), & \text{for } A \subseteq \Omega, A \neq \emptyset. \end{cases} \tag{1.20}$$

This rule boils down to Dempster's rule and disjunctive rule when, respectively, the degree of conflict equals to zero and one. In other cases, it has some intermediate behavior.

Many other alternatives to Dempster's rule can be found in the literature [52,62,97,99, 126], which were mainly proposed to address the issue of combining conflicting information. These alternative rules allow the combination of contradictory information but prevent the representation of certain types of information. In particular, categorical, Bayesian and dogmatic mass functions can often not be properly handled by these alternative combination rules. The use of these alternative rules thus limits the power of belief functions to represent a large variety of information. In this thesis, we adopt the same point of view as Haenni [42] who agreed that the so-called counter-intuitive results that may be obtained from Dempster's rule are often due to erroneous modeling of the pieces of evidence to be combined. Efforts should thus be put on properly representing the information at hand rather than on modifying the combination rule. For instance, in combining unreliable sources of evidence, one can first discount those original pieces of evidence using Shafer's discounting operation [96] (to be introduced in next section) with refined modeling of the reliability. In such a way, the conflict between different pieces of evidence can be greatly reduced and Dempster's rule can be selected as a preferred combination rule.

1.3 Operations over the frame of discernment

In order to manipulate the belief functions more effectively, the conditioning and deconditioning operations are introduced to the framework of belief functions. In addition, we also give an introduction for the discounting operation, which makes it possible to handle partially reliable sources of information.

1.3.1 Conditioning operation

In Bayesian probability theory, conditioning is the fundamental mechanism for updating a probability measure P with new evidence of the form $\omega \in B$ for some $B \subseteq \Omega$ such that $P(B) \neq 0$. The conditional probability measure is defined as

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}, \quad \forall A \subseteq \Omega. \tag{1.21}$$

In a similar way, a *conditioning operation* can also be defined for mass functions in the framework of belief function [101].

Definition 1.7 (conditioning operation)**.** *Let m^Ω be a mass function defined over the frame of discernment Ω. The conditioning of m^Ω with respect to $B \subseteq \Omega$ is defined as*

$$m^\Omega[B] = m^\Omega \oplus m^\Omega_B, \tag{1.22}$$

where m^Ω_B is the categorical mass function with unique focal set B.

From the above definition, we can see that the conditioning operation for mass functions is just a special case of Dempster's rule, in which an arbitrary mass function m^Ω is combined with a categorical mass function m^Ω_B.

1.3.2 Deconditioning operation

In this section, we focus on the inverse question studied in the preceding section. Assume that a source of evidence gives us a mass function representing evidence about some problem defined over the frame Ω, assuming that some proposition $B \subseteq \Omega$ holds. This mass function can be interpreted as a conditional mass function $m^\Omega[B]$ obtained by conditioning some unknown mass function m^Ω by B. However, there will usually exist several mass functions m^Ω verifying this property. In the following, a *deconditioning operation* is introduced to give the least committed mass function, whose conditioning on B yields $m^\Omega[B]$ [101].

Definition 1.8 (deconditioning operation)**.** *Let $m^\Omega[B]$ be a mass function obtained by conditioning some unknown mass function m^Ω by $B \subseteq \Omega$. The deconditioning of $m^\Omega[B]$ with respect to B is defined as*

$$m^\Omega(A) = \begin{cases} m^\Omega[B](C), & \text{if } A = C \cup \overline{B} \text{ for some } C \subseteq \Omega \\ 0, & \text{otherwise,} \end{cases} \tag{1.23}$$

where $\overline{B}$ denotes the complement of set B with respect to set Ω.

Note that the above deconditioning operation usually cannot recover the original unconditioned mass function. It only gives the least committed mass function according to the Least Commitment Principle (LCP) [101], which indicates that, given several mass functions compatible with a set of constraints, the most appropriate is the least informative. In the following, we provide an example to show how the operations of conditioning and deconditioning work to manipulate mass functions.

Exemple 1.4. *We consider the following mass function m defined over the frame $\Omega = \{\omega_1, \omega_2, \omega_3\}$:*

$$m(\{\omega_1\}) = 0.4,\ m(\{\omega_2\}) = 0.3,\ m(\{\omega_3\}) = 0.2,\ m(\Omega) = 0.1.$$

Suppose we know that the truth ω lies in $B = \{\omega_1, \omega_2\}$. With this hypothesis, the above mass function can be updated using the conditioning operation (1.22) with respect to B as:

$$m[B](\{\omega_1\}) = 0.5,\ m[B](\{\omega_2\}) = 3/8,\ m[B](B) = 1/8.$$

Then, with the above conditioned mass function $m[B]$, we intend to recover the original mass function using the deconditioning operation (1.23) with respect to B as:

$$m'(\{\omega_1, \omega_3\}) = 0.5,\ m'(\{\omega_2, \omega_3\}) = 3/8,\ m'(\Omega) = 1/8.$$

We can see that the recovered mass function m' is different from the original mass function m. Therefore, the conditioning and deconditioning operations are not strictly mutually reversible. In addition, the recovered mass function m' becomes less informative than the original mass function m, which indicates that some information is lost in these operations.

1.3.3 Discounting operation

Let us assume that we receive a piece of evidence from a source S, describing some information about the truth ω over the frame of discernment Ω. However, this information is not fully reliable or not fully relevant because, e.g., it is provided by a possible faulty sensor, the measurement was performed in unfavorable experimental condition, or the information is related to a situation that only has some similarity with the situation considered. By considering the information about the reliability of the source, we get a new, less informative mass function. This operation is called *discounting* [96].

Definition 1.9 (discounting operation)**.** *Given a mass function m defined over the frame of discernment Ω and a coefficient $\alpha \in [0, 1]$, the discounting of m with discount rate $1 - \alpha$ yields a new mass function ${}^{\alpha}m$ defined by:*

$${}^{\alpha}m(A) = \begin{cases} \alpha m(A), & \text{for } A \neq \Omega \\ \alpha m(\Omega) + (1 - \alpha), & \text{for } A = \Omega. \end{cases} \tag{1.24}$$

The discounting operation is used to model a situation where a source S provides a mass function m, and the reliability of S is measured by α. If S is fully reliable (i.e., $\alpha = 1$), then m is left unchanged. If S is not reliable at all, m is transformed into the vacuous mass function. One of the effects of the discounting operation is that $Bel(A)$ is reduced and $Pl(A)$ is reinforced or remains unchanged for all $A \subset \Omega$. In other words, the evidence becomes less informative due to the unreliability of the source.

The discounting operation is a very useful tool to build mass function from an unreliable source. Let us reconsider the murder example studied in Example 1.1 and build the mass function with the discounting operation.

Exemple 1.5 (murder continued)**.** *First, we know that a witness saw the murderer and because he is short-sighted, he can only report that he saw a man. Based on this evidence, the following mass function can be constructed for the murderer:*

$$m(\{Peter, John\}) = 1.$$

Further, we know that the witness is drunk 20% *of the time. This means that the above evidence holds with probability* 0.8*. Thus, the discounting operation in Eq. (1.24) with a discount rate of* $1 - \alpha = 0.2$ *can be used to obtain the corresponding mass function as*

$$^{\alpha}m(\{Peter, John\}) = 0.8, \quad ^{\alpha}m(\Omega) = 0.2.$$

Compared with the result in Example 1.1, we can see that the same mass function is built by two different ways. Now, we take a comparison for the information conveyed by the evidence before and after discounting. For the original evidence, both the belief and plausibility of set $\{Peter, John\}$ *equal to* 1*. This means it is certain that the murder is one in* $\{Peter, John\}$*. However, after discounting, the belief of set* $\{Peter, John\}$ *reduces to* 0.8*, while the plausibility keeps unchanged. Accordingly, the support of set* $\{Peter, John\}$ *becomes to an interval* $[0.8, 1]$*, which reflects the uncertainty of the available information.*

1.4 Decision making

In the preceding sections, we have presented some functions and operations that allow us to represent the available uncertain sources of information and to reason with them. The final step is to make a decision about the considered problem based on the reasoning results. In this section, we introduce some decision rules in the framework of belief functions.

1.4.1 Maximum belief/plausibility rule

Let $\Omega = \{\omega_1, \omega_2, \cdots, \omega_n\}$ be the frame of discernment, $\mathcal{A} = \{a_1, a_2, \cdots, a_n\}$ the set of acts, where a_i is the act of selecting ω_i. Define a loss function L: $\mathcal{A} \times \Omega \to \mathbb{R}$ such that

$$L(a_i, \omega_j) = \begin{cases} 0, & \text{if } i = j, \\ 1, & \text{otherwise.} \end{cases}$$

Define the risk of each act a_i as the expected loss if a_i is selected:

$$R_P(a_i) = \mathbb{E}_P[L(a_i, \cdot)] = \sum_{\omega \in \Omega} L(a_i, \omega) P(\{\omega\}), \quad \forall a_i \in \mathcal{A},$$

where P is a probability measure defined on Ω. With the above notions, the decision making problem is formalized to select an act with minimal risk.

In the framework of belief functions, the uncertainty on Ω is described by an interval $[Bel(A), Pl(A)]$, $\forall A \subseteq \Omega$, and the lower and upper expected risk of each act a_i can be defined, respectively, as [27, 98]:

$$\underline{R}(a_i) = \underline{\mathbb{E}}[L(a_i, \cdot)] = \sum_{A \subseteq \Omega} m(A) \min_{\omega \in A} L(a_i, \omega) = 1 - Pl(\{\omega_i\}), \quad \forall a_i \in \mathcal{A},$$
$$\overline{R}(a_i) = \overline{\mathbb{E}}[L(a_i, \cdot)] = \sum_{A \subseteq \Omega} m(A) \max_{\omega \in A} L(a_i, \omega) = 1 - Bel(\{\omega_i\}), \quad \forall a_i \in \mathcal{A}.$$

Minimizing the lower risk $\underline{R}$ and the upper risk $\overline{R}$ result in two decision rules: the *maximum plausibility rule* (optimistic strategy) and the *maximum belief rule* (pessimistic strategy), respectively.

In practice, the maximum plausibility rule is more widely used because it is computationally efficient [11]: suppose $m = m_1 \oplus m_2$, then

$$Pl(\{\omega_i\}) \propto Pl_1(\{\omega_i\}) Pl_2(\{\omega_i\}), \quad \forall \omega_i \in \Omega. \tag{1.25}$$

That is, when combining several mass functions, we do not need to compute the combined mass function using Dempster's rule. Instead, we can compute the combined plausibility using Eq. (1.25) to make the decision according to the maximum plausibility rule.

1.4.2 Maximum pignistic probability rule

In the preceding section, a pessimistic strategy and an optimistic one were derived by minimizing the upper and lower expected risk, respectively. In this section, we introduce another decision rule that finds a compromise between the pessimistic and the optimistic strategies. It is based on the Transferable Belief Model (TBM) [103], which postulates that uncertain reasoning and decision making are two fundamentally different operations occurring at two different levels: uncertain reasoning is performed at the *credal level* using the formalism of belief functions, while decision making is performed at the *pignistic level*, after the mass function has been transformed into a probability measure. This transformation is called the *pignistic transformation*.

Definition 1.10 (pignistic transformation)**.** *Given a mass function m defined over the frame of discernment Ω, the pignistic transformation from the mass function m to a probability measure $BetP$ is defined by:*

$$BetP(A) = \sum_{B \subseteq \Omega} \frac{|A \cap B|}{|B|} m(B), \quad \forall A \subseteq \Omega, \tag{1.26}$$

where $|X|$ denotes the cardinality of set X.

The pignistic probability $BetP(A)$ (from the Latin word *pignus*, meaning a bet) approximates the unknown probability in $[Bel(A), Pl(A)]$:

$$Bel(A) \leq BetP(A) \leq Pl(A), \quad \forall A \subseteq \Omega.$$

Accordingly, the expected risk based on the pignistic probability is

$$R_{BetP}(a_i) = \mathbb{E}_{BetP}[L(a_i, \cdot)] = \sum_{\omega \in \Omega} L(a_i, \omega) BetP(\{\omega\}) = 1 - BetP(\{\omega_i\}), \quad \forall a_i \in \mathcal{A}.$$

Consequently, the expected risk based on the pignistic probability lies in the interval of the lower and upper expected risk:

$$\underline{R}(a_i) \leq R_{BetP}(a_i) \leq \overline{R}(a_i), \quad \forall a_i \in \mathcal{A}.$$

Minimizing the pignistic probability-based risk R_{BetP} results in another decision rule: the *maximum pignistic probability rule*, which is a compromise between the maximum belief rule and the maximum plausibility rule. Next, let us come back to the murder example studied in Example 1.3 and illustrate how these three decision rules work to judge who is the murderer.

Exemple 1.6 (murder continued)**.** *Remember that with the two available pieces of evidence, based on Dempster's rule, we obtain the following combined mass function:*

$$m(\{John\}) = 0.48, \quad m(\{Peter, John\}) = 0.32, \quad m(\{John, Mary\}) = 0.12, \quad m(\Omega) = 0.08,$$

concerning the murderer over the frame $\Omega = \{Peter, John, Mary\}$*. To judge who is the most likely to be the murderer based on the above combined mass function, we compute the belief, plausibility and pignistic probability for each suspect using Eqs. (1.2,1.3,1.26), as shown in Table 1.2. The belief Bel and the plausibility Pl for each suspect provide its lower and upper probabilities to be the murderer, respectively. The pignistic probability BetP provides an approximate estimate between the lower and upper probabilities. For this example the same decision is made by maximizing Bel, Pl and BetP: John is most likely to be the murderer.*

Table 1.2: The belief, plausibility and pignistic probability with regard to each suspect

Suspects	*Bel*	*Pl*	*BetP*
Peter	0	0.4	0.19
John	0.48	1	0.73
Mary	0	0.2	0.09

1.5 Conclusion

As a generalization of probability theory, the theory of belief functions can be used to model and reason with many types of uncertain information. This chapter provided a detailed introduction for the theory of belief functions. Several basic functions that are

commonly used to represent uncertain information have been described. Some combination rules and several other operations (i.e., conditioning, deconditioning and discounting) over the frame of discernment have been introduced. Finally, some decision rules concerning the uncertainty quantified by belief functions have been presented. This theory will be used in the following chapters to model different types of uncertain information encountered in data classification problems.

Classification of uncertain data

Automatic classification of data is an important problem in a variety of engineering and scientific disciplines such as biology, psychology, medicine, marketing, computer vision, military affairs, etc. [50]. In the past several decades, a wide variety of approaches have been developed towards this task. For traditional classification algorithms, the available data are often assumed to be exact or perfect. In many emerging applications, however, the data are inherently uncertain, which brings new challenges to classifier design [1, 67, 81, 86, 112].

In this chapter, we first describe in Section 2.1 the definition of classification and some popular classification methods. In Section 2.2, we discuss the uncertainty in data classification field. Then, we give brief reviews about classification of uncertain data with nearest-neighbor-based approaches in Section 2.3 and rule-based approaches in Section 2.4, respectively. Finally, Section 2.5 concludes this chapter.

2.1 Data classification problem

Before addressing the more advanced issues in data classification field, we first define what we mean by classification, and then introduce some popular classification methods.

2.1.1 What is classification?

The *classification* task occurs in a wide range of human activities. In its wider sense, the term could cover any context in which some decision or forecast is made on the basis of currently available information, and a *classification procedure* is then some formal method for repeatedly making such judgments in new situations [73]. In this thesis we consider a more restricted interpretation. We assume that the problem concerns the construction of a procedure that will be applied to a continuing sequence of cases, in which each new case must be assigned to one of a set of pre-defined *classes* on the basis of observed *features*. The construction of a classification procedure from a set of labeled samples has also been termed *supervised learning* (in order to distinguish it from *unsupervised learning* or *clustering* in which the classes are inferred from the data).

Generally, the classification problem considered in this thesis can be formulated as follows.

- Given
  - a set of M pre-defined classes: $\Omega = \{\omega_1, \cdots, \omega_M\}$, and
  - a set of N labeled training samples: $\mathcal{L} = \{(\mathbf{x}_1, \omega^{(1)}), \cdots, (\mathbf{x}_N, \omega^{(N)})\}$, with each sample described by a feature vector $\mathbf{x} \in \mathbb{R}^P$ and a class label $\omega \in \Omega$,
- the classification problem is to assign a new instance $\mathbf{y} \in \mathbb{R}^P$ to one of the pre-defined classes in Ω based on the available training set $\mathcal{L}$.

2.1.2 Overview of some of the more common classification methods

As a research field closely related to real-world applications, the study of data classification has developed significantly in the past several decades, and a wide variety of approaches have been taken towards this task. According to the used theoretical tools, three main historical strands of research can be identified: *statistical approaches*, *logic-based approaches* and *perceptron-based approaches* [59].

2.1.2.1 Statistical approaches

Statistical approaches are generally characterized by being based on an explicit underlying probability model, which provides a probability of being in each class rather than simply a classification [73]. Under this category of classification approaches, one can find *Bayesian networks*, *k-nearest neighbor*, *support vector machines*, etc.

Bayesian networks [51] A Bayesian Network (BN) is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG). A practical BN commonly used in data classification is the *naive Bayesian* (NB) network which is a very simple BN composed of DAGs with only one parent and several children with a strong assumption of independence among child nodes in the context of their parent. This classifier learns from training data the conditional probability of each feature A_p given the class label ω. Classification is then done by applying Bayes rule to compute the probability of ω given the particular instance of $A_1, \cdots, A_P$, and then predicting the class with the highest posterior probability. The major advantage of the NB classifier is its short computational time for training. However, as the assumption of independence among child nodes is unrealistic for most applications, the classification accuracy of the NB classifier is usually not very high.

k-nearest neighbor [38] The k-nearest neighbor (kNN) rule is based on the principle that the patterns within a data set will generally exist in close proximity to other patterns that have similar properties. With the training samples tagged with class labels, the class label of an unclassified pattern can be determined by observing the class labels of its nearest neighbors. The kNN rule locates the k nearest training samples to the query pattern and determines its class by identifying the single most frequent class label. As a type of lazy learning algorithms, the kNN has been one of the most popular and successful pattern classification techniques due to its simplicity.

Support vector machines [114] The support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, for classification and regression analysis. Given a set of training samples, each marked as belonging to one of two classes, a SVM training algorithm builds a model that assigns new patterns into one class or the other, making it a non-probabilistic binary linear classifier. A SVM model is a representation of the samples as points in space, mapped so that the samples of the separate classes are divided by a clear gap that is as wide as possible. New patterns are then mapped into that same space and predicted to belong to a class based on which side of the gap they fall on. The SVM approach has a sound theoretical foundation, performs well with small datasets, and is insensitive to the number of dimensions.

2.1.2.2 Logic-based approaches

Logic-based approaches aim to mimic the human reasoning process, and to generate classification expressions simple enough to be easily understood [73]. Under this category of classification approaches, one can find *decision trees* and *rule-based classification systems*.

Decision trees [82] A decision tree is a flowchart-like structure in which each internal node represents a test on an feature (e.g., whether a coin flip comes up head or tail), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all features). The paths from root to leaf represent classification rules. First, the training samples are used to build a decision tree, and then new patterns are classified starting at the root node and sorted based on their feature values. One of the most useful characteristics of decision trees is their comprehensibility. People can easily understand why a decision tree classifies a pattern as belonging to a specific class.

Rule-based classification systems [17] Decision trees can be translated into a set of rules by creating a separate rule for each path from the root to a leaf in the tree.

However, rules can also be directly induced from training data using a variety of rule generation algorithms. Generally, a rule-based classification system (RBCS) works in two stages, i.e., to learn a rule base that establishes an association between the feature space and the class space based on the training data, and to classify a query pattern based on the learned rule base with a reasoning method. The RBCS is widely employed in real-world applications due to its capability of building a linguistic model interpretable to users and addressing both quantitative and qualitative information coming from expert knowledge, mathematical models or empirical measures.

2.1.2.3 Perceptron-based approaches

In the data classification field, some well-known algorithms are based on the notion of perceptron [88]. According to the number of perceptron layers used in modeling, these approaches can be further divided as *single-layer perceptrons* and *multi-layer perceptrons*.

Single-layer perceptrons [40] A perceptron is an algorithm for supervised classification of an input into one of several possible non-binary outputs. It is a type of linear classifier, i.e., a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector. The single-layer perceptrons (SLP) algorithm for learning from a batch of training samples iterates until it finds a prediction vector that is correct on all of the training samples. This prediction rule is then used for predicting the labels on the query patterns. The SLP can have lower time complexity when dealing with irrelevant features. However, as a linear classifier, the SLP can only classify linearly separable sets of patterns.

Multi-layer perceptrons [89] A Multi-layer perceptrons (MLP) model has been proposed to solve the nonlinearly separable problem. A MLP consists of large number of units (neurons) joined together in a pattern of network. First, the network is trained on a set of training data to determine input-output mapping. The weights of the connections between neurons are then fixed, and the network is used to determine the class labels of new patterns. MLP has been applied to many real-world problems. However, it is a black-box method: its conclusion cannot be easily interpreted.

2.2 Data classification under uncertainty

In many applications, data contains inherent uncertainty. A number of factors contribute to the uncertainty, such as the random nature of the physical data generation and collection processes, measurement and decision errors, insufficient knowledge, etc. [74]. In this section,

we first provide a brief survey for different categories of uncertainty emerging in real-world applications, and then talk about some issues concerning the uncertainty in data classification field, which are the targets of this thesis.

2.2.1 Types of uncertainty

Uncertainty can be generally characterized as lack of information. Different types of information may be lacking in data and knowledge bases, and consequently give rise to different kinds of uncertainty. As discussed in [21,100], uncertainty can broadly be divided into the following three categories:

- *Incompleteness*: Incomplete information refers to cases where the states of a variable or an event are missing. Consider a database of battlefield reports that should include information about detected unit position (latitude and longitude), category (friendly, neutral, or hostile), and type (tank, armored personnel carrier, or Humvee) for each report. The incomplete information can arise in the following two levels. First, if, for example, the state of the variable representing the type of a detected hostile unit at a specific position is missing from a report, then the information in the report becomes incomplete. Second, if, for example, some reports representing the state of battlefield in one period of time are missing from the database, then the information in the database also becomes incomplete.

- *Imprecision*: Imprecise information refers to cases where the value of a variable is given, but not with enough precision. Suppose the value of the variable representing the type of a detected hostile unit at a certain location is "tank" or "Humvee". The information is complete because the values of each of the three variables representing the unit's position, category, and type are given, but the information is imprecise because there is some ambiguity as to the exact type of the detected unit.

- *Unreliability*: Unreliable information refers to cases where the information is complete, precise, but uncertain since it might be wrong. This type of uncertainty appears when the observer (human or sensor) is taken into account. It is the observer that is not reliable about the available information.

The above descriptions broadly categorize different types of uncertainty according to the types of information lacking in data and knowledge bases. This partition is not unique and each type of uncertainty can be further subdivided into several subtypes. One can refer to Chapter 3 of [21] for more information.

2.2.2 Uncertainty in data classification

Different types of uncertainty described in the preceding section exist widely in data classification problems. In this thesis, we confine to the following four critical issues concerning the uncertainty in data classification field.

1. *Imprecise information in overlapping regions*: For most real-world classification problems, the patterns from different classes usually partly overlap. Though the training samples in overlapping regions are assigned with precise labels, they actually cannot be seen as truly representatives of their corresponding clusters.

2. *Incomplete training data set*: Incomplete training data set refers to cases where the training data set is non-exhaustive. Generally, if the training data set is exhaustive enough to characterize the real class-conditional probability distributions, data classification is easy to do for any classifier. However, in small data set situations, the ideal behaviors of many traditional classifiers degrade dramatically.

3. *Unreliable training data set*: Unreliable training data set refers to cases where the training data have noisy class labels or feature values. The class noise, also known as labeling error, occurs when a sample is assigned to an incorrect class. In contrast, the feature noise is used to refer to corruptions in the values of one or more features of samples in a data set.

4. *Partial training data and expert knowledge*: In some real-world classification problems, both partial training data collected by sensors and partial expert knowledge provided by human may be available. These two types of information are usually independent but complementary for classification.

As reviewed in Chapter 1, the theory of belief functions is an effective tool to model and reason with uncertain information. The use of belief function theory in pattern classification fields is not new, and some classifiers have already been developed based on belief functions in the past. For instance, Smets [101] and Appriou [7] have proposed model-based classifiers based on the generalized Bayes theorem (GBT) [101] which is an extension of Bayes theorem in transferable belief model (TBM) [103]. There are some other case-based evidential classifiers based on k-nearest neighbors [26, 68, 69, 137], support vector machines [61], decision trees [113], and neural network [28]. In essence, these methods are devoted to addressing different types of uncertainty based on different classification models.

In this thesis, we aim to solve the above listed four uncertain data classification problems in the framework of belief functions. As pointed out in the title, our work mainly focuses on two popular classification approaches: nearest-neighbor-based classification and rule-based classification. Specifically, we study the first two issues in nearest-neighbor-based

classification, and the last two issues in rule-based classification. We will give a review of related work in the following two sections.

2.3 Nearest-neighbor-based classification

In classification problems, complete statistical knowledge regarding the conditional density of each class is rarely available, which precludes applications of the optimal Bayes classification procedure. In these cases, a good solution is to classify each new pattern using the evidence of nearby sample observations. One such non-parametric procedure has been introduced by Fix and Hodges [38], and has since become well-known in the pattern recognition field as the k-nearest neighbor (kNN) rule. It is one of those algorithms that are very simple to understand but work quite well in practice. In this section, we first describe the main principle of the kNN rule, and then emphasize on two issues that affect the performance of the kNN rule.

2.3.1 k-nearest neighbor rule

The kNN rule is based on a simple method of density estimation. The idea is very similar to kernel density estimation [109]. Instead of using kernel functions, here the estimation is made in a simple way. For estimating the density at a point $\mathbf{x}$, place a hypercube centered at $\mathbf{x}$ and keep increasing its size till k neighbors are captured. Then the estimate of the density at point $\mathbf{x}$ is given by

$$\hat{p}(\mathbf{x}) = \frac{k}{NV}, \tag{2.1}$$

where N is the number of total samples and V is the volume of the hypercube.

Having obtained an expression for a density estimate, we can now use this in a decision rule. Suppose that in the first k samples there are k_m samples from class ω_m (so that $\sum_{m=1}^{M} k_m = k$). Let the total number of samples in class ω_m be N_m (so that $\sum_{m=1}^{M} N_m = N$). Then, the class-conditional density can be estimated as

$$\hat{p}(\mathbf{x} \mid \omega_m) = \frac{k_m}{N_m V}, \tag{2.2}$$

and the prior probability as

$$\hat{p}(\omega_m) = \frac{N_m}{N}. \tag{2.3}$$

Then the decision rule is to assign x to ω_m, if

$$\hat{p}(\omega_m \mid \mathbf{x}) \geq \hat{p}(\omega_i \mid \mathbf{x}), \quad \text{for all } i = 1, 2, \cdots, M, \tag{2.4}$$

or, using Bayes' theorem,

$$\frac{k_m}{N_m V} \frac{N_m}{N} \geq \frac{k_i}{N_i V} \frac{N_i}{N}, \quad \text{for all } i = 1, 2, \cdots, M, \tag{2.5}$$

that is, assign x to ω_m, if

$$k_m \geq k_i, \quad \text{for all } i = 1, 2, \cdots, M. \tag{2.6}$$

Thus, the decision rule is to assign x to the class that receives the largest vote amongst the k nearest neighbors.

The kNN rule is an non-parametric lazy learning algorithm. First, it is non-parametric, which means that it does not make any assumptions on the underlying data distribution. This is quite useful, as in the real world, most of the practical data does not obey the typical theoretical assumptions made (e.g., Gaussian mixtures, linearly separable, etc.). Second, it is a lazy algorithm, which means that there is no explicit training phase, or it is very minimal. In addition, Cover and Hart [20] have provided a statistical justification of this procedure by showing that, as the numbers N and k both tend to infinity in such a way that $k/N \to 0$, the error rate of the kNN rule approaches the optimal Bayes error rate. Beyond these remarkable properties, the kNN rule owes much of its popularity in the pattern recognition community to its good performance in practical applications.

However, in the finite sample case, the classical k-NN rule is not guaranteed to be the optimal way of using the information contained in the neighborhood of unclassified patterns. This is the reason why the improvement of this rule has remained an active research topic in the past 60 years. There are several key issues that affect the performance of the kNN rule. One of the problems encountered in using the voting kNN rule is that the distances from different nearest neighbors are neglected in the decision. To address this problem, several variants have been proposed [26, 36, 75]. Particularly, an evidential version of k-nearest neighbor rule (EkNN) has been proposed based on the theory of belief functions in [26]. In EkNN, each neighbor of a sample to be classified is considered as an item of evidence supporting certain hypotheses concerning the class membership of that sample. The evidence of the k nearest neighbors is then pooled by means of Dempster's rule of combination. This approach provides a well treatment for the uncertainty caused by the distance issue.

In this thesis, we focus on another two major issues in the kNN rule. One is that in using the kNN rule, each of the training samples is considered equally important in the assignment of the class label to the query pattern. This limitation frequently causes difficulty in regions where the data sets from different classes overlap. Atypical samples in overlapping regions are given as much weight as those that are truly representatives of the clusters. In order to overcome this difficulty, the sample editing procedure [122] was proposed to preprocess the original training samples. Besides, the choice of the distance metric [130] is another important consideration, especially in small data set situations. Although various metrics can be used to compute the distance between two points, the most desirable distance metric is one for which a smaller distance between two samples

implies a greater likelihood of having the same class. In the following two sections, we will provide brief reviews for the existing sample editing methods and distance metrics.

2.3.2 Sample editing methods

In order to address the uncertain information in overlapping regions, several editing procedures have been proposed to preprocess the original training samples [55,58,76,110,115,122]. According to the structure of the edited labels, the editing procedures can be divided into two categories: crisp editing and soft editing.

In [122], Wilson proposed a simple editing procedure to preprocess the training set. This procedure classifies a training sample $\mathbf{x}_i$ using the kNN rule with the remainder of the training set, and deletes it from the original training set if its original label $\omega^{(i)}$ does not agree with the classification result. Later, concerned with the possibility of large amounts of samples being removed from the training set, Koplowitz and Brown [58] developed a modification of the simple editing technique. For a given value of k, another parameter k' is defined such that $(k+1)/2 \leq k' \leq k$. Instead of deleting all of the conflicting samples, if a particular class (excluding the original class) has at least k' representatives among these k nearest neighbors, then $\mathbf{x}_i$ is labeled according to that majority class (see Algorithm 1). Essentially, both the simple editing procedure and its modification belong to the category of crisp editing procedures, in which each edited sample is either removed or assigned to a single class.

Algorithm 1: Modified simple editing algorithm

Require: the original training set $\mathcal{T}$ composed of N labeled samples, two parameters k and k' with $(k+1)/2 \leq k' \leq k$
$\mathcal{T}' \leftarrow \mathcal{T}$
for $i = 1$ to N **do**
  Find k nearest neighbors of $\mathbf{x}_i$ in $\mathcal{T} \setminus \{\mathbf{x}_i, \omega^{(i)}\}$
  if a class label, say c, is held by at least k' neighbors **then**
    set the label of $\mathbf{x}_i$ in $\mathcal{T}'$ to c
  else
    remove $\{\mathbf{x}_i, \omega^{(i)}\}$ from $\mathcal{T}'$
  end if
end for
return the edited training set $\mathcal{T}'$

In order to overcome the weakness of the crisp editing method, a fuzzy editing procedure was proposed that reassigns fuzzy membership to each training sample $\mathbf{x}_i$ based on its k_{edit}

nearest neighbors according to the following equation [55]:

$$u_j(\mathbf{x}_i) = \begin{cases} 0.51 + (k_j/k_{edit}) * 0.49, & \text{if } \omega_j = \omega^{(i)} \\ (k_j/k_{edit}) * 0.49, & \text{if } \omega_j \neq \omega^{(i)}, \end{cases} \tag{2.7}$$

where the value k_j is the number of the neighbors found which belong to class ω_j. This fuzzy editing procedure belongs to the soft editing category, in which each edited sample can be assigned to several classes. It provides more detailed information about the samples' membership than the crisp editing procedures.

2.3.3 Distance metrics

As the core of the kNN rule, the distance metric plays a crucial role in determining the classification performance, especially in small data set situations. To overcome the limitations of the original Euclidean (L2) distance metric, a number of methods have been proposed. According to the structure of the metric, these methods can be mainly divided into two categories: global distance metric learning [10, 37, 119, 124], and local distance metric learning [49, 79, 80, 118, 121, 131].

The global distance metric learning approach learns the distance metric in a global sense, i.e., the same global weighted (GW) distance metric is defined for all of the patterns:

$$d_{GW}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^{P} \lambda_j^2 (x_j - y_j)^2}, \tag{2.8}$$

where $\mathbf{x}$ is a sample in the training set, $\mathbf{y}$ is a query pattern to be classified, and λ_j is the weight of the j-th feature. Although the above global distance metric is intuitively appealing, it is too coarse as the feature weights of the distance metric are irrelevant with the class labels of the patterns. For classification problems with a large number of classes, it is hard to learn a GW distance metric that can simultaneously separate all of the class pairs well.

In contrast, the local distance metric learning approach can learn a local distance metric for some specific patterns. According to the types of the used local information, this approach can be further subdivided into two categories: geometry-based local distance metric learning and label-based local distance metric learning. For the geometry-based local distance metric learning, the aim is to learn a locally adaptive distance metric in the neighborhood of each query pattern. Recently, Paredes et al. [79, 80] provided another idea to learn the locally adaptive distance metric, i.e., the local distance metric is relevant to the class labels of the training samples. In their work, a class-dependent weighted (CDW) distance metric was defined as

$$d_{CDW}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^{P} \lambda_c^{j\,2} (x_j - y_j)^2}, \tag{2.9}$$

where λ_c^j is the weight of the j-th feature and c is the class index of training sample $\mathbf{x}$. The CDW distance metric provides more freedom than the GW distance metric and can be learnt adaptively for different classes of the training samples. However, as illustrated in the following example, this distance metric is insufficient to reflect the local specificities in feature space for query patterns from different classes.

Exemple 2.1. *Figure 2.1 illustrates a simple three-class classification problem, where the data in each class are uniformly distributed.* $(\mathbf{x}_1, A)$, $(\mathbf{x}_2, B)$ *and* $(\mathbf{x}_3, C)$ *are two-dimensional data points in training set* $\mathcal{T}$. $\mathbf{y}_1$ *and* $\mathbf{y}_2$ *are the query data to be classified. Considering the classification of data* $\mathbf{y}_1$ *between Class A and Class B, when calculating the distance of* $\mathbf{y}_1$ *to* $\mathbf{x}_1$ *and* $\mathbf{x}_2$*, intuitively, to avoid classifying it as Class B mistakenly, feature X should be assigned a larger weight. However, when classifying data* $\mathbf{y}_2$ *as Class B or Class C, feature Y should be assigned a larger weight to determine the distance of* $\mathbf{y}_2$ *to* $\mathbf{x}_2$ *and* $\mathbf{x}_3$*. However, as indicated in Eq. (2.9), the CDW distance is only relevant to the class labels of the training samples, it lacks the flexibility of designing local feature weights for query patterns from different classes.*

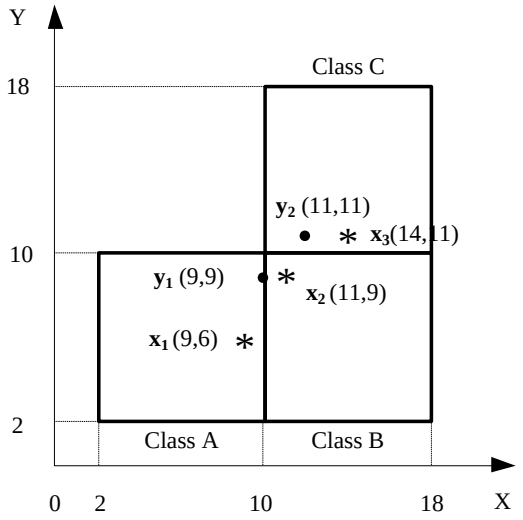


Figure 2.1: A three-class classification example

2.4 Rule-based classification

Rules are one of the most common forms for representing various kinds of knowledge. Rule-based systems, usually constructed from human knowledge in forms of IF-THEN rules, are often applied to inference problems [14]. However, for most classification problems, the available information is in the format of a collection of training data. If rules can be learnt from the training data, the rule-based systems can be used for classification purpose. One such learning procedure has been introduced by Chi et al. [17], and has since become

well-known in the pattern recognition field as the fuzzy rule-based classification system (FRBCS). Due to its capability of building linguistic models interpretable to users, the FRBCS has been successfully applied to different real-world classification tasks, including image processing [104], intrusion detection [111], fault classification [93], and medical applications [3, 116]. In this section, we first describe the main principle of the FRBCS developed by Chi et al. [17], and then provide a brief review for different improved FRBCSs in order to get better accuracy and robustness. Finally, we discuss the possibility and necessity of combining partial training data and expert knowledge based on the rule-based systems to perform classification.

2.4.1 Fuzzy rule-based classification system

A fuzzy rule-based classification system (FRBCS) is composed of two main conceptual components, the fuzzy rule base (FRB) and the fuzzy reasoning method (FRM). The FRB establishes an association between the space of pattern features and the space of consequent classes. The FRM provides a mechanism to classify a query pattern based on the FRB.

The fuzzy rule in the FRB for an M-class (denoted as $\Omega = \{\omega_1, \omega_2, \cdots, \omega_M\}$) pattern classification problem with P features has the following structure [17]:

$$\begin{aligned} \text{Fuzzy Rule } R^q: \quad & \text{If } x_1 \text{ is } A_1^q \text{ and } \cdots \text{ and } x_P \text{ is } A_P^q, \text{ then consequence is } C^q \\ & \text{with rule weight } \theta^q, \quad q = 1, 2, \cdots, Q, \end{aligned} \tag{2.10}$$

where $\mathbf{x} = (x_1, x_2, \cdots, x_P)$ is the pattern feature vector and $\mathbf{A}^q = (A_1^q, \cdots, A_P^q)$ is the antecedent part, with each A_p^q belonging to fuzzy partitions $\{A_{p,1}, A_{p,2}, \cdots, A_{p,n_p}\}$ associated with the p-th feature. $C^q \in \Omega$ is the label of the consequent class, and Q is the number of fuzzy rules in the FBR. The rule weight θ^q, characterizing the certainty grade of the fuzzy rule R^q, is used as the strength of R^q in fuzzy reasoning.

Based on the above fuzzy rule structure, several FRB generation methods have been proposed [16, 17, 47]. Here, we introduce the one proposed by Chi et al. [17] because it is one of the most widely used algorithms. To generate the FRB, this method uses the following steps:

1. *Construction of the fuzzy regions.* Usually, the partition of the pattern space is related to the specific classification problem. If no prior knowledge is available, the method based on fuzzy grids is usually employed [46]. Figure 2.2 shows an example of the fuzzy partition of a two-dimensional pattern space with triangular fuzzy sets. Based on this method, once the domain interval and the partition number for each feature are determined, the fuzzy regions are easily computed.

2. *Generation of a fuzzy rule for each training pattern.* Assume that N labeled P-dimensional training patterns $\mathbf{x}_i = (x_{i1}, \cdots, x_{iP})$, $i = 1, 2, \cdots, N$ are available. For

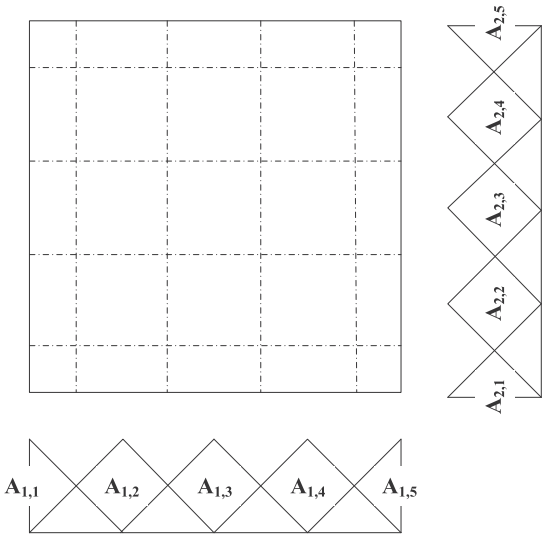


Figure 2.2: An example of the fuzzy partition of a two-dimensional pattern space by fuzzy grids

each training pattern $\mathbf{x}_i$, the following steps are necessary:

(a) To calculate its matching degree $\mu(\mathbf{x}_i)$ with different fuzzy regions using the geometric mean operator

$$\mu_{\mathbf{A}^q}(\mathbf{x}_i) = \sqrt[P]{\prod_{p=1}^{P} \mu_{A_p^q}(x_{ip})}, \tag{2.11}$$

where $\mu_{A_p^q}(\cdot)$ is the membership function of the fuzzy set A_p^q;

(b) To assign the training pattern $\mathbf{x}_i$ to the fuzzy region with the greatest matching degree;

(c) To generate a rule for this training pattern, with the antecedent part determined by the selected fuzzy region, the consequent class determined by the class label of the training pattern, and the rule weight determined by the greatest matching degree.

3. *Reduction of the generated FRB.* Rules with the same antecedent part may be generated during the learning process. In this case, only the one having the maximum rule weight is maintained, and the other ones are removed.

Once the FRB has been constructed, the query patterns can be classified by the following single winner FRM [17]. Let S be the set of Q constructed fuzzy rules. A query pattern $\mathbf{y} = (y_1, y_2, \cdots, y_P)$ is classified by a single winner rule R^w, which is chosen from rule set S as

$$R^w = \arg\max_{R^q \in S} \{\mu_{\mathbf{A}^q}(\mathbf{y})\theta^q\}. \tag{2.12}$$

That is, the winner rule R^w has the maximum product of the matching degree $\mu_{\mathbf{A}^q}(\mathbf{y})$ and the rule weight θ^q in S. Because it is limited by the number of training patterns, the query pattern may not always be covered by any rule in the FRB, in which case the classification is rejected. To avoid the non-covering problem, several techniques have been proposed, such as using bell-shaped fuzzy sets instead of the triangular fuzzy sets [78] or stretching a rule by deleting one or more of its antecedent terms [45].

As one of the computational intelligence-based classification approaches, the most useful characteristic of the FRBCS is high interpretability. In contrast with statistical machine learning approaches, which are based on complicated statistical models and perceptron-based approaches, which are based on black-box models, the FRBCS can build a linguistic model interpretable to users. In addition, due to the use of fuzzy sets, the FRBCS can use both quantitative and qualitative information coming from expert knowledge, mathematical models or empirical measures [95].

However, the FRBCS may face lack of accuracy when dealing with some complex applications, due to the inflexibility of the concept of the linguistic variable, which imposes hard restrictions on the fuzzy rule structure [5]. For example, when the input-output mapping varies in complexity within the space, homogeneous partitioning using linguistic variables for the input and output spaces becomes inefficient. Besides, the fuzzy rule structure is not robust to pattern noise and wrong rules may be generated in noisy conditions, which hinders its applications in harsh working conditions (e.g., battlefield target recognition). Therefore, plenty of work has been done in the past two decades in order to improve the accuracy and robustness of the FRBCS. In the following section, we will give a brief review of different improved FRBCSs.

2.4.2 Improved FRBCSs

In the past two decades, many researchers have proposed variants of the original FRBCS developed by Chi et al. [17], in order to improve its accuracy and robustness. These approaches can be mainly divided into two categories: learning an optimized rule base with sophisticated methods, and changing the rule structure to make it more flexible to characterize the input-output mapping.

Approaches in the first category are motivated by the fact that unsatisfactory classification results may be due to defects of the rule generation method. The original fuzzy rule structure displayed as Eq. (2.10) is still used, but optimized rule bases are learnt with sophisticated methods. For example, in [16], the rule base was built from the given training samples based on a support vector learning approach; in [4, 18], the genetic algorithms were used for designing fuzzy rule bases and determining an appropriate combination of antecedent and consequent linguistic values of each fuzzy rule.

In contrast, work in the second category is mainly based on the claim that the original fuzzy rule structure is insufficient to characterize the complex input-output mapping. The original fuzzy rule structure is extended in different ways to become more flexible (e.g., by extending the consequent part or the antecedent part). In [19], the following fuzzy rule structure with certainty degrees for all classes in the consequent part was provided:

$$R^q: \text{ If } x_1 \text{ is } A_1^q \text{ and } \cdots \text{ and } x_P \text{ is } A_P^q, \text{ then } (\theta_1^q, \cdots, \theta_M^q), \quad q = 1, 2, \cdots, Q, \tag{2.13}$$

where θ_j^q is the certainty degree for rule R^q to predict class ω_j for a pattern belonging to the fuzzy region represented by the antecedent part of the rule. This type of rule extends the original fuzzy rule using different values for the consequent part $(\theta_1^q, \cdots, \theta_M^q)$. Considering

$$\theta_j^q = \begin{cases} \theta^q, & \text{if } \omega_j = C^q \\ 0, & \text{otherwise,} \end{cases}$$

we get the original fuzzy rule. Accordingly, based on this new extended fuzzy rule structure, an additive combination reasoning method is employed to classify a query pattern $\mathbf{y}$ as follows:

$$\omega = \arg\max_{\omega_j \in \Omega} \sum_{q=1}^{Q} \mu_{\mathbf{A}^q}(\mathbf{y}) \cdot \theta_j^q. \tag{2.14}$$

A different approach was proposed in [64], where the antecedent part of the original fuzzy rule is extended with belief degrees embedded in each antecedent terms as follows:

$$\begin{aligned} R^q: \quad & \text{If } x_1 \text{ is } \{(A_{1,j}, \alpha_{1,j}^q)\}_{j=1}^{n_1} \text{ and } \cdots \text{ and } x_P \text{ is } \{(A_{P,j}, \alpha_{P,j}^q)\}_{j=1}^{n_P}, \\ & \text{then consequence is } C^q, \text{ with rule weight } \theta^q, \quad q = 1, 2, \cdots, Q, \end{aligned} \tag{2.15}$$

where the antecedent term for each feature $\{(A_{p,j}, \alpha_{p,j}^q)\}_{j=1}^{n_p}$ is in belief distribution. Considering

$$\alpha_{p,j}^q = \begin{cases} 1, & \text{if } A_{p,j} = A_p^q \\ 0, & \text{otherwise} \end{cases}, \quad p = 1, \cdots, P,$$

we get the original fuzzy rule. Accordingly, based on this rule structure, each training sample is developed as a rule to model the input-output relationship. The query pattern is then classified by the additive combination of the weighted consequences of all of the generated rules.

2.4.3 Classification with partial training data and expert knowledge

According to the type of information used in modeling, pattern classification methods can be categorized into data-driven and knowledge-driven [107]. Data-driven models are based on learning from the training data characterizing the problem at hand. These data are usually collected by sensors observing the environment. In the previous part of this chapter, we mainly focused on data-driven models. In contrast, knowledge-driven models are based

on the expert knowledge understanding a particular domain or problem. Expert knowledge is often encoded into rules or relations, based on which an inference is made [70]. Examples of the most popular systems using knowledge-driven models are expert systems [48,90] and decision support systems [13, 44].

In some real-world pattern classification applications, both training data and expert knowledge may be available. For example, for target recognition [9,23], historical measurements from a long period of collection by sensors can be used to train the target recognition system. In addition, the expert knowledge about target characteristics obtained from the manufacturers or intelligence also provides important information for target recognition. In the classification process, training data and expert knowledge are usually complementary. The training data trends to provide a relatively fine estimate for the real class-conditional distribution, but they may be unreliable in some specific regions of feature space, due to limited training patterns and the potential measurement noise. In contrast, the expert knowledge usually provides a relatively rough but overall reliable estimate for the real class-conditional distribution.

In the past, several methods have been proposed to address the classification problem based on both training data and expert knowledge. For instance, Zhou et al. [134] proposed to build an initial rule base from the available expert knowledge, and then to optimize the rule base by adding or pruning rules based on training data. In [108], Tang et al. developed a knowledge-based naive Bayes classifier, which uses training data to estimate the involved conditional probabilities. Later, in [107], they proposed another similar method, which builds a fuzzy rule-based system based on expert knowledge and then uses training data to optimize the involved fuzzy membership functions. In essence, these methods follow the same idea that first building a base model from expert knowledge and then optimizing this model based on training data. However, one major disadvantage of this idea is that the weights of training data and expert knowledge cannot be adjusted according to the qualities of these two types of information. To overcome this limitation, we intend to address the hybrid classification problem with a different idea. That is, to build a data-driven model and a knowledge-driven model independently, and then to combine them into an adaptive hybrid classification model by taking into account their weights.

In order to integrate training data and expert knowledge for classification, we need to find a common representation model that can make use of both the two types of information. The IF-THEN rule is a good representation model because, on the one hand, as reviewed in the previous section, the IF-THEN rules can be learnt from training data and, on the other hand, expert knowledge is also easily coded into IF-THEN rules. Many rule-based systems have been proposed to deal with the classification problems using either training data or expert knowledge [46,84,94,111]. Therefore, it is a natural way to combine partial training data and expert knowledge based on the rule-based systems.

2.5 Conclusion

Different types of uncertainty may exist in real-world data classification problems. In this chapter, we discussed four critical issues concerning the uncertainty in data classification field, i.e., imprecise information in overlapping regions, incomplete training data set, unreliable training data set, and incomplete training data and expert knowledge, which are the targets of this thesis. In the following four chapters, we will solve the above uncertain data classification problems in the framework of belief functions based on two classification approaches: nearest-neighbor-based classification and rule-based classification.

Part II

Nearest-neighbor-based classification

This part focuses on classification of uncertain data using nearest-neighbor-based approaches.

Chapter 3 focuses on improving the performance of the kNN rule for cases in which patterns from different classes overlap strongly. An evidential editing version of the kNN rule is developed based on the theory of belief functions in order to well model the imprecise information for those samples in overlapping regions.

Chapter 4 concerns the classification problems based on incomplete training data sets. A polychotomous classification problem is solved by combining a group of locally learned pairwise kNN classifiers in the framework of belief functions to deal with the uncertain output information.

Evidential editing k-nearest neighbor classifier

One of the difficulties that arise when using the k-nearest neighbor rule is that each of the labeled training samples is given equal importance in deciding the class of the query pattern to be classified, regardless of their typicality. In this chapter, the theory of belief functions is introduced into the k-nearest neighbor rule to develop an evidential editing version of this algorithm. An evidential editing procedure is proposed to reassign the original training samples with new labels represented by an evidential membership structure. With the introduction of the evidential editing procedure, the imprecise information in overlapping regions can be well characterized. After the evidential editing, a classification procedure is developed to handle the more general situation in which the edited training samples are assigned dependent evidential labels.

In this chapter, we first describe in Section 3.1 the background and motivations. The details of the proposed evidential editing k-nearest neighbor classifier are presented in Section 3.2. Four experiments are performed in Section 3.3 to evaluate the performance of the proposed method. Finally, Section 3.4 concludes this chapter.

3.1 Introduction

As one of the most well-known classification methods, the k-nearest neighbor (kNN) rule [38] provides a simple non-parametric procedure for the assignment of a class label to the query pattern based on its k nearest neighbors. In this decision rule, the class label represented by each neighbor is considered equally important, regardless of their typicality. This rule may have difficulty for cases where the data sets from different classes overlap strongly. It may be argued that the atypical samples in overlapping regions should not be given as much weight as those that are truly representatives of the clusters. In order to overcome this difficulty, the editing procedure was proposed to preprocess the original training samples and the kNN rule was used to classify the query pattern based on the edited training samples.

As reviewed in Section 2.3.2, according to the structure of the edited labels, the editing procedures can be divided into crisp editing and soft editing. A fuzzy editing procedure, proposed by Keller et al. [55], as a type of soft editing techniques, can provide more detailed information about the sample membership than the crisp editing procedures. However, different types of uncertainty may coexist in real-world classification problems, e.g., fuzziness may coexist with imprecision. The fuzzy editing procedure, which is based on fuzzy set theory [132], cannot address imprecise information effectively in the modeling and reasoning processes. In contrast, the theory of belief functions can well model imprecise information thanks to the belief functions defined on the power set of the frame of discernment. The theory of belief functions has already been used in the kNN rule [26,31,68,69,137]. However, these methods mainly focus on modeling the uncertainty in the classification process, none of them considers any editing procedure and the original training set is used to make classification.

In this chapter, an evidential editing k-nearest neighbor (EEkNN) is proposed based on the theory of belief functions. The proposed EEkNN classifier contains two stages: evidential editing and classification. First, an evidential editing procedure reassigns the original training samples with new labels represented by an evidential membership structure. Compared with the fuzzy membership used in fuzzy editing, the evidential labels provide more expressiveness to characterize the imprecision for those samples in overlapping regions. After the evidential editing procedure, a classification procedure is developed to classify a query pattern based on the edited training samples.

3.2 Evidential editing k-nearest neighbor classifier

Let us consider an M-class classification problem and let $\Omega = \{\omega_1, \cdots, \omega_M\}$ be the set of classes. Assuming that a set of N labeled training samples $\mathcal{T} = \{(\mathbf{x}_1, \omega^{(1)}), \cdots, (\mathbf{x}_N, \omega^{(N)})\}$ with input vectors $\mathbf{x}_i \in \mathbb{R}^P$ and class labels $\omega^{(i)} \in \Omega$ are available, the problem is to classify a query pattern $\mathbf{y} \in \mathbb{R}^P$ based on the training set $\mathcal{T}$.

The proposed evidential editing k-nearest neighbor (EEkNN) procedure is composed of the following two stages:

1. *Preprocessing (evidential editing)*: The evidential editing algorithm assigns evidential labels to each labeled sample.

2. *Classification*: The class of the query pattern is decided based on the distance to the sample's k nearest neighbors and these k nearest neighbors evidential membership information.

3.2.1 Evidential editing

The goal of the evidential editing stage is to assign to each sample in the training set $\mathcal{T}$ a new soft label with an evidential structure as follows:

$$\mathcal{T}' = \{(\mathbf{x}_1, m_1), (\mathbf{x}_2, m_2), \cdots, (\mathbf{x}_N, m_N)\}, \tag{3.1}$$

where m_i, $i = 1, 2, \cdots, N$, are mass functions defined on the frame of discernment Ω.

Remark 3.1. *The above evidential membership structure provides a more general representation model than the traditional crisp label or fuzzy membership. For one training sample $\mathbf{x}_i$, if there is no imprecision among the frame of discernment Ω, the evidential membership just reduces to fuzzy membership as a special case. Further, if there is also no probability uncertainty, it finally reduces to crisp label.*

The problem is now to compute an evidential label for each training sample. In [26], an evidential k-nearest neighbor (EkNN) rule was proposed based on the theory of belief functions, where the classification result of the query pattern is a mass function. In the following part, we use the EkNN rule to carry out the evidential editing.

For each training sample $\mathbf{x}_i$, $i = 1, 2, \cdots, N$, we denote the leave-it-out training set as $\mathcal{T}_i = \mathcal{T} \setminus \{(\mathbf{x}_i, \omega^{(i)})\}$, $i = 1, 2, \cdots, N$. Now, we consider the evidential editing for one training sample $\mathbf{x}_i$ on the basis of the information contained in $\mathcal{T}_i$. For the training sample $\mathbf{x}_i$, each neighbor $\mathbf{x}_j$ $(j \neq i)$ provides an item of evidence regarding the class membership of $\mathbf{x}_i$ as follows

$$\begin{cases} m_i(\{\omega^q\} \mid \mathbf{x}_j) & = \alpha\phi_q(d_{ij}) \\ m_i(\Omega \mid \mathbf{x}_j) & = 1 - \alpha\phi_q(d_{ij}) \\ m_i(A \mid \mathbf{x}_j) & = 0, \quad \forall A \in 2^\Omega \setminus \{\Omega, \{\omega_q\}\}, \end{cases} \tag{3.2}$$

where $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$, ω_q is the class label of $\mathbf{x}_j$ (that is, $\omega^{(j)} = \omega_q$), and α is a parameter such that $0 < \alpha < 1$. As suggested in [26], $\alpha = 0.95$ can be used to obtain good results on average. When d is the Euclidean distance, a good choice for ϕ_q is

$$\phi_q(d) = \exp(-\gamma_q d^2), \tag{3.3}$$

where γ_q is a positive parameter associated to class ω_q, which can be heuristically set to the inverse of the mean squared Euclidean distance between training samples belonging to class ω_q.

Based on the distance $d(\mathbf{x}_i, \mathbf{x}_j)$, we select the k_{edit} nearest neighbors of $\mathbf{x}_i$ in training set $\mathcal{T}_i$ and calculate the corresponding k_{edit} mass functions in the above way. As the items of evidence from different neighbors are independent (because the training samples are usually measured or collected independently), the k_{edit} mass functions are combined using

Dempster's rule defined by Eq. (1.10) to form a resulting mass function m_i, synthesizing the final evidential membership regarding the label of $\mathbf{x}_i$ as

$$m_i = m_i(\cdot \mid \mathbf{x}_{i_1}) \oplus m_i(\cdot \mid \mathbf{x}_{i_2}) \oplus \cdots \oplus m_i(\cdot \mid \mathbf{x}_{i_{k_{edit}}}), \tag{3.4}$$

where $i_1, i_2, \cdots, i_{k_{edit}}$ are the indices of the k_{edit} nearest neighbors of $\mathbf{x}_i$ in $\mathcal{T}_i$. Generally, the selection of parameter k_{edit} depends on the specific classification problem. In practice, we can use cross-validation to search for the best value.

3.2.2 Classification

After the evidential editing procedure introduced in Section 3.2.1, the problem now turns into classifying a query pattern $\mathbf{y} \in \mathbb{R}^P$ based on the new edited training set $\mathcal{T}'$ as shown in Eq. (3.1). In this section, we extend the evidential k-nearest neighbor (EkNN) rule [26] to handle the more general situation in which the edited training samples are assigned dependent evidential labels. This classification procedure is composed of the following two steps: first, the mass functions from the k nearest neighbors of the query pattern are computed; then, the k mass functions are combined to obtain the final result.

3.2.2.1 Determination of the mass functions

Consider the k nearest neighbors of the query pattern $\mathbf{y}$. If one training sample $\mathbf{x}_i$ is very close to $\mathbf{y}$, generally, it means that $\mathbf{x}_i$ is a very reliable piece of evidence for the classification of $\mathbf{y}$. In contrast, if $\mathbf{x}_i$ if far from $\mathbf{y}$, then it provides only little reliable evidence. In the theory of belief functions, Shafer's discounting operation defined by Eq. (1.24) can be used to discount the unreliable evidence before combination.

Denote as m_i the class membership of the training sample $\mathbf{x}_i$, and β_i the confidence degree of the class membership of $\mathbf{y}$ with respect to the training sample $\mathbf{x}_i$. The evidence provided by $\mathbf{x}_i$ for the class membership of $\mathbf{y}$ is represented with a discounted mass function ${}^{\beta_i}m_i$ by discounting m_i with a discount rate $1 - \beta_i$ as follows:

$$\begin{cases} {}^{\beta_i}m_i(\{\omega_q\}) & = \beta_i m_i(\{\omega_q\}), \quad q = 1, 2, \cdots, M \\ {}^{\beta_i}m_i(\Omega) & = \beta_i m_i(\Omega) + (1 - \beta_i). \end{cases} \tag{3.5}$$

The confidence degree β_i is determined based on the distance d_i between $\mathbf{x}_i$ and $\mathbf{y}$, in such a way that a larger distance results in a smaller confidence degree. Thus, β_i should be a decreasing function of d_i. We use a similar decreasing function with Eq. (3.3) to define the confidence degree $\beta_i \in (0, 1]$ as

$$\beta_i = \exp(-\lambda_i d_i^2), \tag{3.6}$$

where λ_i is a positive parameter associated to the training sample $\mathbf{x}_i$ and is defined as

$$\lambda_i = \left[\sum_{q=1}^{M} m_i(\{\omega_q\}) \overline{d}^q + m_i(\Omega) \overline{d} \right]^{-2}, \tag{3.7}$$

where $\overline{d}$ is the mean distance between all training samples, and $\overline{d}^q$ is the mean distance between training samples belonging to each class ω_q, $q = 1, 2, \cdots, M$.

Remark 3.2. *In calculating the confidence degree, parameter λ_i is designed by extending the parameter γ_q in Eq. (3.3) to the evidential membership structure. In Eq. (3.7), suppose that the label of the training sample $\mathbf{x}_i$ is crisp with ω_q, i.e., $m_i(\{\omega_q\}) = 1$, $m_i(\{\omega_j\}) = 0$, for $j = 1, 2, \cdots, M$, $j \neq q$, $m_i(\Omega) = 0$. Then, the parameter λ_i reduces to γ_q in the case of crisp labels. With this design for parameter λ_i, the confidence degree β_i defined in Eq. (3.6) is not independent with the mass function m_i any longer. Considering that the dependence between them is quite weak (because distance d_i is more influential than m_i in determining β_i), we still use Shafer's discounting operation to obtain the discounted mass function approximately.*

3.2.2.2 Combination of the mass functions

To make a decision about the class of the query pattern $\mathbf{y}$, the generated k mass functions should be combined to obtain the final fusion result. For combination, Dempster's rule relies on the assumption that the items of evidence combined are independent. However, as illustrated in the following example, in the editing process, common training samples may be used for calculating the class membership of different edited samples. Therefore, the items of evidence from different edited samples to classify the query pattern $\mathbf{y}$ cannot be regarded as independent.

Exemple 3.1. *Figure 3.1 illustrates the dependence between different edited training samples. In this example, "△" denotes the training samples, and "□" denotes the query pattern. In the evidential editing process, k_{edit} is set to 2 to search for the nearest neighbors, and in the classification process, the number of nearest neighbors k is set to 3. The three nearest neighbors used for the classification of the query pattern $\mathbf{y}$ are $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$. As, in the evidential editing process, the training sample $\mathbf{x}_4$ is used for calculating both the class membership of $\mathbf{x}_1$ and $\mathbf{x}_2$, the edited training samples $\mathbf{x}_1$ and $\mathbf{x}_2$ are no longer independent. In contrast, the edited training samples $\mathbf{x}_3$ is still independent with both $\mathbf{x}_1$ and $\mathbf{x}_2$ as they did not use common training samples in the evidential editing process. Therefore, the items of evidence from different edited training samples to classify the query pattern $\mathbf{y}$ may have partial dependence.*

To account for this dependence, we use the parameterized t-norm based combination

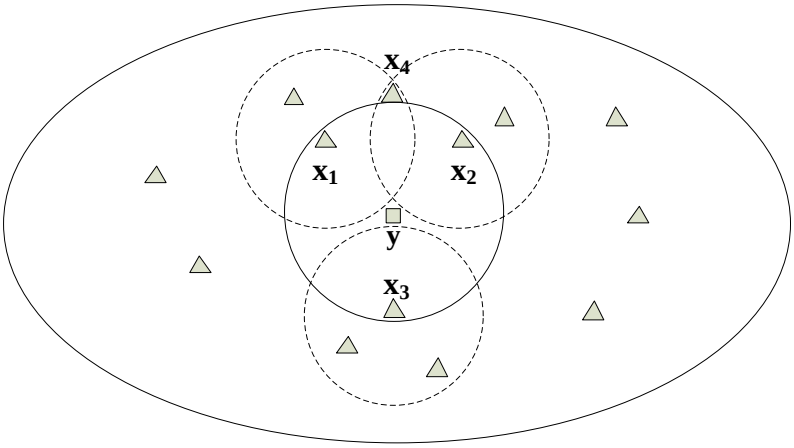


Figure 3.1: Illustration of dependence between edited training samples

rule shown in Eq. (1.18) and obtain the final combination result for query pattern $\mathbf{y}$ as

$$m = {}^{\beta_{i_1}} m_{i_1} \circledast_s {}^{\beta_{i_2}} m_{i_2} \circledast_s \cdots \circledast_s {}^{\beta_{i_k}} m_{i_k}, \tag{3.8}$$

where $i_1, i_2, \cdots, i_k$ are the indices of the k nearest neighbors of $\mathbf{y}$ in $\mathcal{T}'$. The selection of parameter s depends on the potential dependence degrees of the edited samples. In practice, we can use the cross-validation to search for the optimal t-norms based combination rule.

For making decisions based on the above combined mass function m, the pignistic probability $BetP$ shown in Eq. (1.26) is used and the query pattern $\mathbf{y}$ is assigned to the class with the maximum pignistic probability.

3.3 Experiments

The performance of the proposed evidential editing k-nearest neighbor (EEkNN) classifier was evaluated in four different experiments. In the first experiment, the combination rule used in the EEkNN classifier was evaluated under different dependence degrees of the edited samples. In the second experiment, the effects of the two main parameters k_{edit} and k in the EEkNN classifier were analyzed. In the last two experiments, the performance of the EEkNN classifier was compared with those of other nearest-neighbor-based methods (the modified simple editing kNN (SEkNN) [58], the fuzzy editing kNN (FEkNN) [55] and the evidential kNN (EkNN) [26]) using synthetic data sets and real data sets.

3.3.1 Evaluation of the combination rules

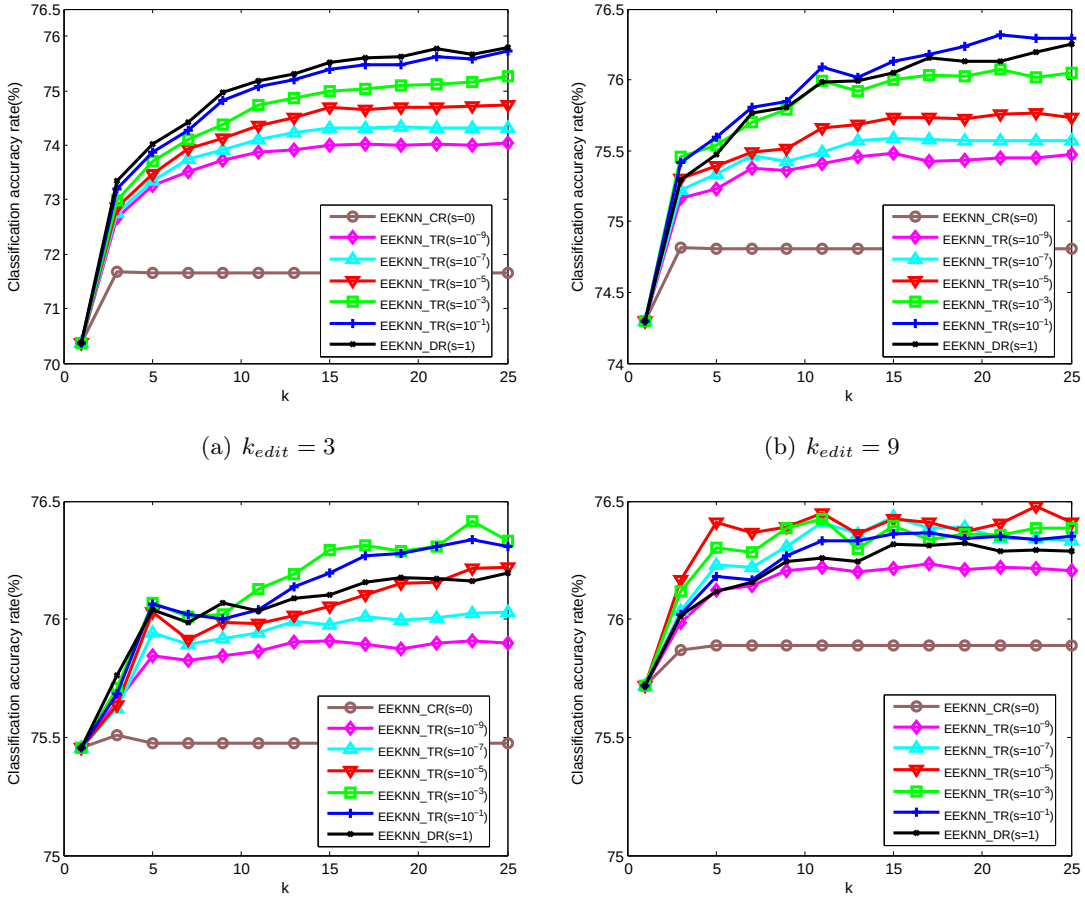
This experiment was designed to evaluate the combination rules used in the proposed EEkNN method. A two-dimensional three-class classification problem was considered. The following class-conditional normal distributions were assumed:

Class A: $\mu_A = (6, 6)^T$, $\Sigma_A = 4\mathbf{I}$;

Class B: $\mu_B = (14, 6)^T$, $\Sigma_B = 4\mathbf{I}$;

Class C: $\mu_C = (14, 14)^T$, $\Sigma_C = 4\mathbf{I}$.

A training set of 150 samples and a test set of 3000 samples were generated from the above distributions using equal prior probabilities. For each case, 30 trials were performed with 30 independent training sets. The average test classification rate over the 30 trials was calculated. In the preprocessing stage, $k_{edit} = 3, 9, 15, 21$ were selected. For classification, values of k ranging from 1 to 25 have been investigated. The t-norms based combination rules (TR) with parameter s ranging from 0 to 1 have been evaluated. Note that the cautious rule (CR) is retrieved when $s = 0$, and the Dempster's rule (DR) is retrieved when $s = 1$.



(a) $k_{edit} = 3$ (b) $k_{edit} = 9$

(c) $k_{edit} = 15$ (d) $k_{edit} = 21$

Figure 3.2: Classification results for different combination rules and different k_{edit} values with values of k ranging from 1 to 25

Figure 3.2 shows the classification accuracy for different combination rules and different k_{edit} values with values of k ranging from 1 to 25. It can be seen that the best combination

rules vary with changes of the value of k_{edit}. In other words, the k_{edit} value has great influence to the dependence of the edited samples. A larger k_{edit} value tends to result in larger dependence. For one specific classification problem, the selection of the best combination rule depends on the potential dependence of the edited samples, which further depends on the utilized k_{edit} value. Therefore, for the EEkNN method, the optimal t-norms based combination rule should be searched for each specific k_{edit} value.

3.3.2 Parameter analysis

This experiment was designed to analyze the effect of parameters k_{edit} and k for the proposed EEkNN method. The same training and test samples with the previous experiment were used. The difference is that in the preprocessing stage, $k_{edit} = 3, 6, 9, 12, 15, 18, 21, 24$ were selected and the optimal t-norms based combination rule for each specific k_{edit} value was used to make the classification. Average classification accuracy over the 30 trials with values of k ranging from 1 to 25 has been investigated.

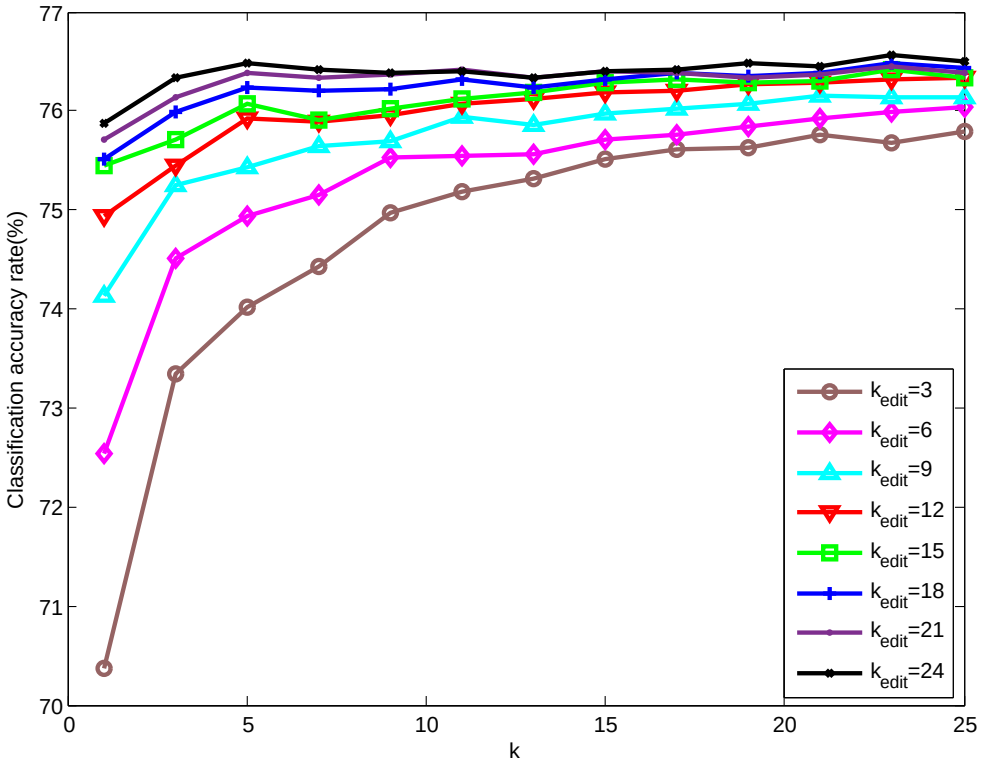


Figure 3.3: Classification results of the EEkNN method for different k_{edit} and k

From Figure 3.3, it can be seen that the classification performance can improve clearly as the parameter k_{edit} increases within an interval ([3,12] in this example). When k_{edit} exceeds an upper boundary ($\overline{k_{edit}} = 12$ in this example), the classification performance no longer improves ideally. In addition, when k_{edit} takes small values, the classification

performance can improve as the parameter k increases. Whereas, when k_{edit} exceeds the upper boundary, the parameter k has little effect to the classification performance.

3.3.3 Synthetic data test

This experiment was designed to compare the proposed EEkNN with other nearest-neighbor-based methods using synthetic data sets with different class overlapping ratios, defined as the number of training samples in the overlapping region divided by the total number of training samples. A training sample $\mathbf{x}_i$ is considered to be in the overlapping region if its corresponding maximum plausibility $Pl_i^{\max}$ after evidential editing is less than a set upper bound Pl^*, namely, $Pl^* = 0.9$. A two-dimensional three-class classification problem was considered. The following class-conditional normal distributions were assumed. For comparisons, we changed the variance of each distribution to control the class overlapping ratio.

Case 1 Class A: $\mu_A = (6,6)^T, \Sigma_A = 3\mathbf{I}$; Class B: $\mu_B = (14,6)^T, \Sigma_B = 3\mathbf{I}$;
Class C: $\mu_C = (14,14)^T, \Sigma_C = 3\mathbf{I}$. Overlapping ratio $\rho = 6.67\%$

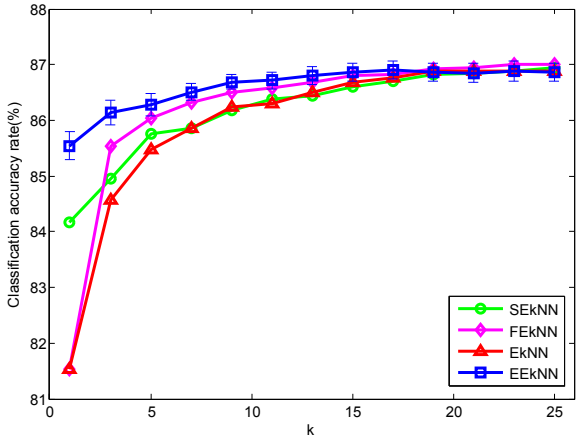
Case 2 Class A: $\mu_A = (6,6)^T, \Sigma_A = 4\mathbf{I}$; Class B: $\mu_B = (14,6)^T, \Sigma_B = 4\mathbf{I}$;
Class C: $\mu_C = (14,14)^T, \Sigma_C = 4\mathbf{I}$. Overlapping ratio $\rho = 10.00\%$

Case 3 Class A: $\mu_A = (6,6)^T, \Sigma_A = 5\mathbf{I}$; Class B: $\mu_B = (14,6)^T, \Sigma_B = 5\mathbf{I}$;
Class C: $\mu_C = (14,14)^T, \Sigma_C = 5\mathbf{I}$. Overlapping ratio $\rho = 21.33\%$

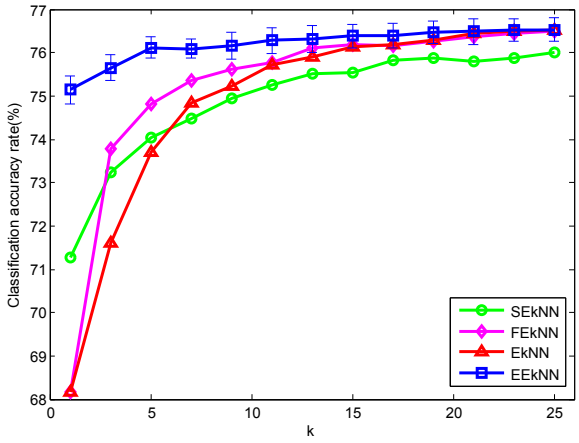
A training set of 150 samples and a test set of 3000 samples were generated from the above distributions using equal prior probabilities. For each case, 30 trials were performed with 30 independent training sets. The average classification accuracy and the corresponding 95% confidence interval[1] were calculated. For each trial, the best values for the parameters k_{edit} and s in the EEkNN method were determined in the sets $\{3,6,9,12,15,18,21,24\}$ and $\{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 0\}$, respectively, by cross-validation. For all of the considered method, values of k ranging from 1 to 25 have been investigated.

Figure 3.4 shows the classification results for synthetic data sets with different overlapping ratios. It can be seen that, for the three cases, the EEkNN method provides better classification performance than other nearest-neighbor-based methods. With the increase of the class overlapping ratio, the performance improvement becomes more important. Furthermore, the EEkNN method is less sensitive to the value of k and it performs well even with a small value of k.

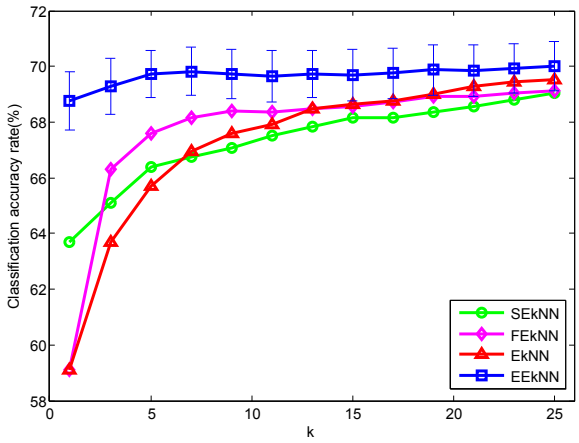
[1] Computed as $\left[\overline{p} - u_{1-\frac{\alpha}{2}}\frac{S}{\sqrt{N_t}}, \overline{p} + u_{1-\frac{\alpha}{2}}\frac{S}{\sqrt{N_t}}\right]$, where $\overline{p} = \frac{1}{N_t}\sum_{i=1}^{N_t} p_i$, $S = \sqrt{\frac{1}{N_t - 1}\sum_{i=1}^{N_t}(p_i - \overline{p})^2}$, with p_i, $i = 1, 2, \cdots, N_t$, being the classification accuracy rates for the N_t trials.



(a) Case 1: $\rho = 6.67\%$



(b) Case 2: $\rho = 10.00\%$



(c) Case 3: $\rho = 21.33\%$

Figure 3.4: Classification results for synthetic data sets with different overlapping ratios (SEkNN: modified simple editing kNN, FEkNN: fuzzy editing kNN, EkNN: evidential kNN, EEkNN: evidential editing kNN)

3.3.4 Real data test

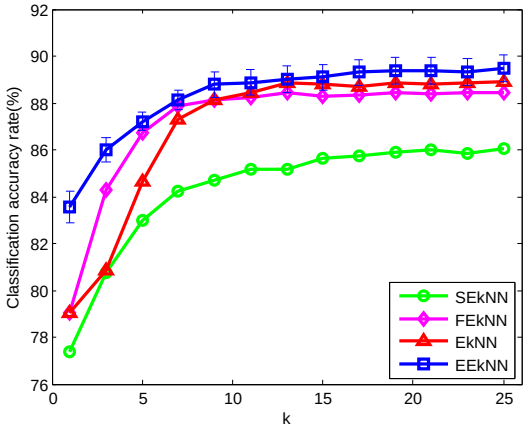
This experiment was designed to compare the proposed EEkNN with other nearest-neighbor-based methods using some real-world classification problems from the well-known UCI Repository of Machine Learning Databases [72]. The main characteristics of the 10 real data sets used in this experiment are summarized in Table 3.1. To assess the results, we considered the resampled paired test. A series of 30 trials was conducted. In each trials, the available samples were randomly divided into a training set and a test set (with equal sizes). For each data set, we calculated the average classification rate of the 30 trials and the corresponding 95% confidence interval. For the proposed EEkNN method, the best values for the parameters k_{edit} and s were determined with the same procedure used in the previous experiment. For all of the considered method, values of k ranging from 1 to 25 have been investigated.

Table 3.1: Description of the benchmark data sets employed in the study

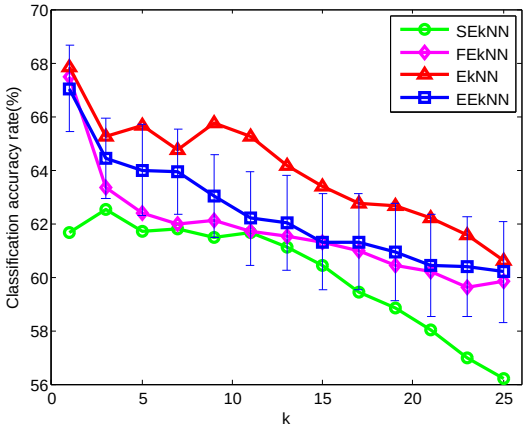
Data set	# Instances	# Features	# Classes	Overlapping ratio
Balance	625	4	3	19.23%
Glass	214	9	6	11.04%
Haberman	306	3	2	18.59%
Ionosphere	214	9	6	18.29%
Liver	345	6	2	19.19%
Pima[a]	336	8	2	19.05%
Transfusion	748	4	2	20.60%
Vertebral	310	6	3	11.20%
Waveform	5,000	21	3	19.60%
Yeast	1,484	8	10	22.74%

[a]For the data sets containing missing values, instances with missing feature values are removed.

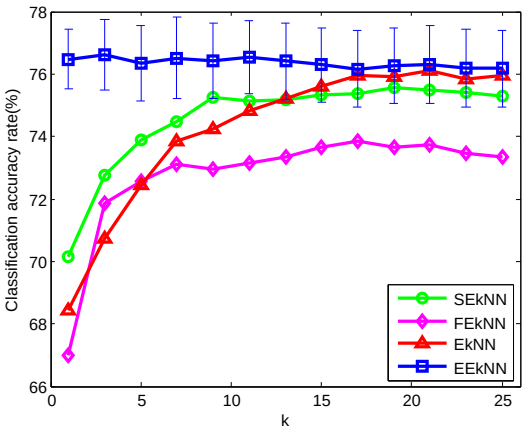
Figure 3.5 shows the classification results of different methods for benchmark data sets. It can be seen that, for data sets with high overlapping ratios, such as *Balance*, *Haberman*, *Ionosphere*, *Liver*, *Pima*, *Transfusion*, *Waveform* and *Yeast*, the EEkNN method provides better classification performance than other nearest-neighbor-based methods, especially for small value of k. In contrast, for those data sets with relatively low overlapping ratios, such as *Glass* and *Vertebral*, the classification performances of different methods were quite similar. The reason is that, for these two data sets, the best classification performance was obtained when k took a small value and, under this circumstance, the evidential editing cannot improve the classification performance. We can also see that, different from other nearest-neighbor based methods, the EEkNN method is less sensitive to the value of k and it usually performs well even with a small value of k.



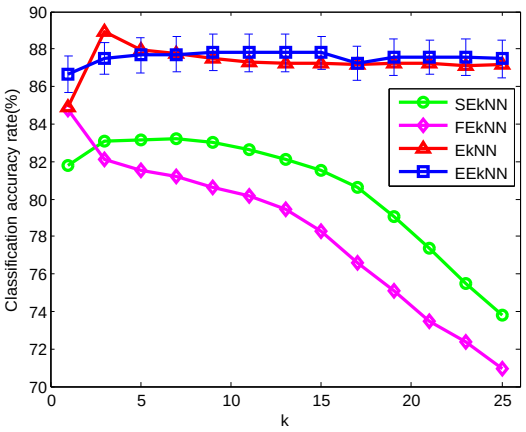
(a) Balance



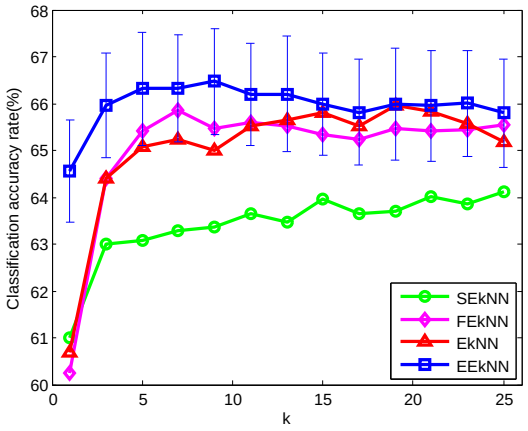
(b) Glass



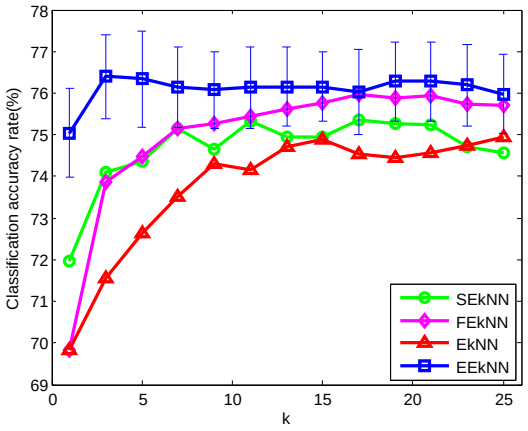
(c) Haberman



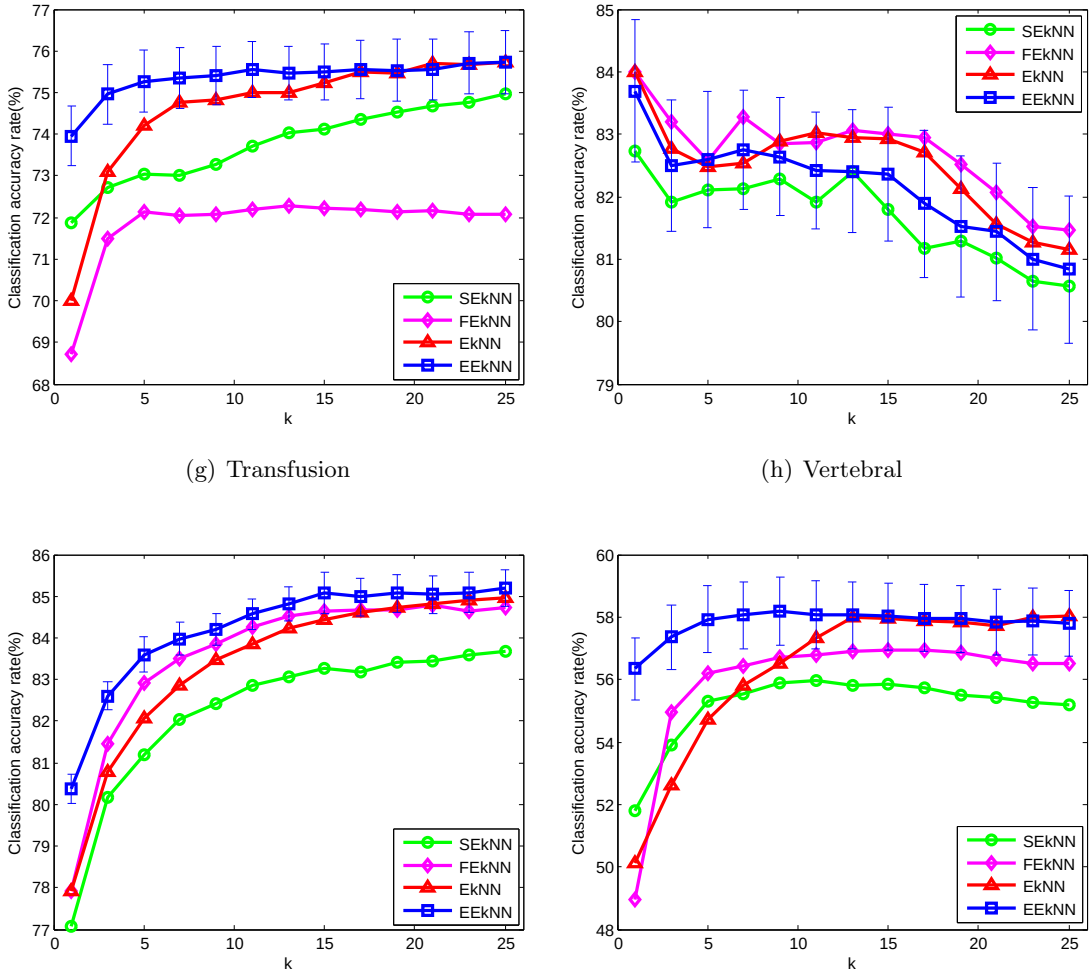
(d) Ionosphere



(e) Liver



(f) Pima



(g) Transfusion (h) Vertebral

(i) Waveform (j) Yeast

Figure 3.5: Classification results of different methods for benchmark data sets

3.4 Conclusion

An evidential editing k-nearest neighbor (EEkNN) classifier has been developed based on an evidential editing procedure that reassigns the original training samples with new labels represented by an evidential membership structure. Thanks to this procedure, patterns situated in overlapping regions have less influence on the decisions. Our results show that the proposed EEkNN classifier achieves better performance than other considered nearest-neighbor-based methods, especially for data sets with high overlapping ratios. In particular, the proposed EEkNN classifier is not too sensitive to the value of k and it can gain a quite good performance even with a small value of k. This is an advantage in time or space-critical applications, in which only a small value of k is permitted in the classification process.

Evidential fusion of pairwise k-nearest neighbor classifiers

The performance of the k-nearest neighbor (kNN) classifier is known to be very sensitive to the distance metric used in classifying a query pattern, especially in small training data set cases. In this chapter, a pairwise distance metric related to pairs of class labels is proposed. Compared with the existing distance metrics, it provides greater flexibility to design the feature weights so that the local specificities in feature space can be well characterized. Based on the proposed pairwise distance metric, a polychotomous classification problem is solved by combining a group of pairwise kNN (PkNN) classifiers in the framework of belief functions to deal with the uncertain output information.

In this chapter, we first describe the background and motivations in Section 4.1. In Section 4.2, a pairwise distance metric is defined and a parameter optimization procedure is designed based on maximum likelihood principle. Using the proposed pairwise distance metric, the corresponding PkNN classifiers are combined in the framework of belief functions in Section 4.3. Two experiments to evaluate the performance of the proposed method are reported in Section 4.4. Finally, Section 4.5 concludes this chapter.

4.1 Introduction

The k-nearest neighbor (kNN) rule is known to have good performance for large training data set [20]. However, in many practical pattern classification applications, the training data set is incomplete, and the real class-conditional probability distributions cannot be well characterized using the limited training samples. In such small training data set situations (relative to the intrinsic dimensionality of the data involved), the ideal asymptotical behavior of the kNN classifier degrades dramatically [33]. This observation motivates the growing interest in finding variants of the kNN rule and adequate distance metrics that potentially improve the kNN classification performance in small data set situations.

As the core of the kNN rule, the distance metric plays a crucial role in determining

the classification performance. To overcome the limitations of the original Euclidean (L2) distance metric, a number of methods have been proposed to address the distance metric learning issue. As reviewed in Section 2.3.3, according to the structure of the metric, these methods can be mainly divided into global distance metric learning and local distance metric learning. The main drawback of the global learning approach is that the learned single distance metric usually cannot separates all of the class pairs well. As one representative label-based local distance metric learning method, Paredes et al. [79] proposed to learn a class-dependent weighted (CDW) distance metric adaptively for each class. However, as illustrated in Example 2.1, as the learned CDW distance metric is only relevant to the class labels of the training samples, it is insufficient to reflect the local specificities in feature space for query patterns in different classes.

In this chapter, we focus on the label-based local distance metric learning problem. To overcome the limitations of the CDW distance metric, a pairwise distance metric related to the labels of the class pairs to be classified is defined. For a polychotomous classification problem, instead of learning a global distance metric, we decompose it into learning a group of pairwise distance metrics. Because only two classes are involved for each pairwise distance metric, the feature weights can be learnt in a more local way. Based on each learned pairwise distance metric, a pairwise kNN (PkNN) classifier can be designed to separate two classes. Then, a polychotomous kNN classification problem can be solved by fusing several PkNN classifiers. A variety of schemes have been proposed for deriving a combined decision from individual ones, such as voting rule [39], Bayes combination [60], multilayered perceptrons [105], etc. Considering that the output of each PkNN classifier may have high uncertainty, the PkNN classifiers are combined in the framework of belief functions due to its well capability of modeling and combining uncertain information.

4.2 Pairwise distance metric learning

To better characterize the local specificities in feature space, in Section 4.2.1, we define a pairwise weighted distance metric and design a parameter optimization procedure to learn it based on the maximum likelihood principle. Then, in Section 4.2.2, we extend the pairwise weighted distance metric to further consider the potential correlation between different features.

4.2.1 Pairwise weighted distance metric learning

Definition 4.1 (Pairwise weighted distance metric)**.** *Suppose* $\mathbf{x}$ *and* $\mathbf{y}$ *are two* P*-dimensional patterns whose labels belong to class pair* $\Omega_{p,q} = \{\omega_p, \omega_q\}$*. The pairwise weighted (PW)*

distance metric between $\mathbf{x}$ *and* $\mathbf{y}$ *is defined as*

$$d_{PW}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{j=1}^{P} {\lambda_{p,q}^{j}}^2 (x_j - y_j)^2}, \tag{4.1}$$

where $\lambda_{p,q}^{j}$ *is a constant that weights the role of the* j*-th feature in the distance metric concerning class pair* $\Omega_{p,q}$.

This definition includes, as particular cases, the distance metrics reviewed in Section 2.3.3. If $\lambda_{p,q}^{j} = 1$ for all $p = 1, \cdots, M$, $q = 1, \cdots, M$, $j = 1, \cdots, P$, the above defined PW distance metric reduces to the L2 distance metric. In addition, the GW and CDW distance metrics correspond to the cases where the metric weights do not depend on the class labels or are only dependent on the class label of the first pattern, respectively. Therefore, the PW distance metric provides a more general dissimilarity measure than the L2, GW or CDW distance metrics.

Remark 4.1. *Compared with the the above mentioned distance metrics, the PW distance metric provides greater flexibility to design the feature weights so that the local specificities in feature space can be well characterized. We study the three-class classification problem illustrated in Example 2.1 again. In Figure 4.1, using the PW distance metric, to discriminate Class B and Class A,* $\lambda_{B,A}^{X}$ *(the two subscripts denote the class labels, the superscript denotes the feature index) can take much larger value than* $\lambda_{B,A}^{Y}$*. In contrast, one can assign smaller value to* $\lambda_{B,C}^{X}$ *than to* $\lambda_{B,C}^{Y}$ *to discriminate Class B and Class C.*

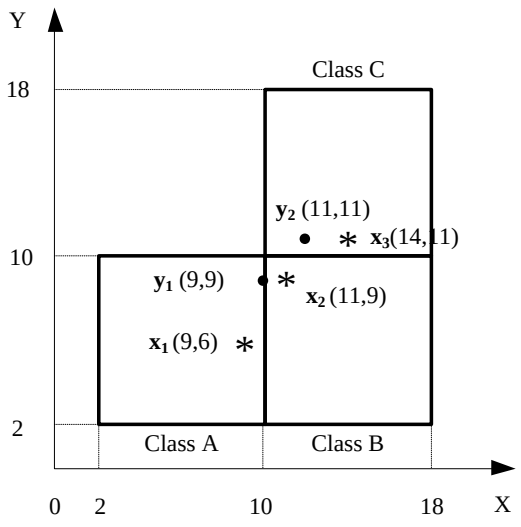


Figure 4.1: A three-class classification example

In the above defined PW distance metric, the only free parameters are the feature weights related to the labels of the two considered classes. In the following part, we aim to learn feature weights $\lambda_{p,q}^{j}$ $(1 \le p < q \le M,\ j = 1, \cdots, P)$ from the training data by

optimizing some criteria. A simple way of defining the criteria for the desired metric is to keep the data pairs from the same class close to each other while separating those data pairs from different classes far from each other [124].

We divide training set $\mathcal{T}$ into M subsets $\mathcal{T}_k$, $k = 1, \cdots, M$, with each $\mathcal{T}_k$ containing all of the N_k training data belonging to class ω_k:

$$\mathcal{T}_k = \{(\mathbf{x}_i, \omega_k) \mid i \in I_k\},$$

where I_k is the set of indices for training data $\mathbf{x}_i$ belonging to class ω_k.

We now consider learning feature weights $\lambda_{p,q}^j$ $(j = 1, \cdots, P)$ from training subsets $\mathcal{T}_p$ and $\mathcal{T}_q$. Let us denote the set of data pairs from the same class as

$$\mathcal{S} = \{(\mathbf{x}_m, \mathbf{x}_n) \mid m, n \in I_p; m < n\} \cup \{(\mathbf{x}_m, \mathbf{x}_n) \mid m, n \in I_q; m < n\},$$

and the set of data pairs from different classes as

$$\mathcal{D} = \{(\mathbf{x}_m, \mathbf{x}_n) \mid m \in I_p; n \in I_q\}.$$

Following the idea presented in [130], a logistic regression model can be assumed when estimating the probability for any data pair $(\mathbf{x}_m, \mathbf{x}_n)$ to share the same class,

$$\Pr(+ \mid (\mathbf{x}_m, \mathbf{x}_n)) = \frac{1}{1 + \exp\left(d_{PW}^2(\mathbf{x}_m, \mathbf{x}_n) - \mu_{p,q}\right)}, \tag{4.2}$$

and then the probability for any data pair $(\mathbf{x}_m, \mathbf{x}_n)$ to share different classes is

$$\begin{aligned} \Pr(- \mid (\mathbf{x}_m, \mathbf{x}_n)) &= 1 - \frac{1}{1 + \exp\left(d_{PW}^2(\mathbf{x}_m, \mathbf{x}_n) - \mu_{p,q}\right)} \\ &= \frac{1}{1 + \exp\left(-d_{PW}^2(\mathbf{x}_m, \mathbf{x}_n) + \mu_{p,q}\right)}, \end{aligned} \tag{4.3}$$

where "+" and "−" denote data pair $(\mathbf{x}_m, \mathbf{x}_n)$ belonging to the same class and different classes, respectively. Parameter $\mu_{p,q}$ is the threshold. The data pair $(\mathbf{x}_m, \mathbf{x}_n)$ will be assigned higher probability to be in the same class when their square PW distance is much smaller than threshold $\mu_{p,q}$. In contrast, if their square PW distance is much larger than threshold $\mu_{p,q}$, they will be given more probability to have different classes.

Then, the overall log-likelihood for both the data pairs in $\mathcal{S}$ and $\mathcal{D}$ can be written as

$$\begin{aligned} \mathcal{L}_g\left(\{\lambda_{p,q}^j\}_{j=1}^P, \mu_{p,q}\right) &= \log \Pr(+ \mid \mathcal{S}) + \log \Pr(- \mid \mathcal{D}) \\ &= -\sum_{(\mathbf{x}_m, \mathbf{x}_n) \in \mathcal{S}} \log\left(1 + \exp\left(d_{PW}^2(\mathbf{x}_m, \mathbf{x}_n) - \mu_{p,q}\right)\right) \\ &\quad -\sum_{(\mathbf{x}_m, \mathbf{x}_n) \in \mathcal{D}} \log\left(1 + \exp\left(-d_{PW}^2(\mathbf{x}_m, \mathbf{x}_n) + \mu_{p,q}\right)\right) \\ &= -\sum_{(\mathbf{x}_m, \mathbf{x}_n) \in \mathcal{S}} \log\left(1 + \exp\left(\sum_{j=1}^P {\lambda_{p,q}^j}^2 (x_{mj} - x_{nj})^2 - \mu_{p,q}\right)\right) \\ &\quad -\sum_{(\mathbf{x}_m, \mathbf{x}_n) \in \mathcal{D}} \log\left(1 + \exp\left(-\sum_{j=1}^P {\lambda_{p,q}^j}^2 (x_{mj} - x_{nj})^2 + \mu_{p,q}\right)\right). \end{aligned} \tag{4.4}$$

With the maximum likelihood estimation, the PW distance metric learning can be formulated as the following optimization problem

$$\begin{array}{ll} \max\limits_{\{\lambda_{p,q}^j\}_{j=1}^P, \mu_{p,q}} & \mathcal{L}_g\left(\{\lambda_{p,q}^j\}_{j=1}^P, \mu_{p,q}\right) \\ \text{s.t.} & \lambda_{p,q}^j \geq 0,\ j = 1, \cdots, P, \text{ and } \mu_{p,q} \geq 0. \end{array} \tag{4.5}$$

This is a convex programming problem, which can be solved using Newton's method [8].

4.2.2 Extension to pairwise Mahalanobis distance metric learning

In the previous section, the distance metric was learnt under the assumption that the P considered features are independent. However, in many real-world applications, this assumption is hardly tenable [120]. Therefore, in the following, we extend the PW distance metric to further consider the correlation between different features.

Definition 4.2 (Pairwise Mahalanobis distance metric)**.** *Suppose* $\mathbf{x}$ *and* $\mathbf{y}$ *are two* P*-dimensional patterns whose labels belong to class pair* $\Omega_{p,q} = \{\omega_p, \omega_q\}$*. The pairwise Mahalanobis (PM) distance metric between* $\mathbf{x}$ *and* $\mathbf{y}$ *is defined as*

$$d_{PM}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \mathbf{A}_{p,q} (\mathbf{x} - \mathbf{y})}, \tag{4.6}$$

where $\mathbf{A}_{p,q} \in \mathbb{R}^{P \times P}$ *is a positive semi-definite matrix (i.e.,* $\mathbf{A}_{p,q} \succeq 0$*) that weights the role of features in the distance metric concerning class pair* $\Omega_{p,q}$*. If we restrict* $\mathbf{A}_{p,q}$ *to be diagonal, the defined PM distance metric reduces to the PW distance metric.*

In a similar way as described in the previous section, the PM distance metric learning can also be formulated as a nonlinear optimization problem. However, in the case of learning a full matrix $\mathbf{A}_{p,q}$, the constraint that $\mathbf{A}_{p,q}$ be positive semi-definite becomes difficult to enforce, and Newton's method often becomes prohibitively expensive (requiring $O(P^6)$ time to invert the Hessian over P^2 parameters). To simplify the computation, we model the matrix $\mathbf{A}_{p,q}$ using the eigenspace of the training samples [130]. Based on training subsets $\mathcal{T}_p$ and $\mathcal{T}_q$, the covariance matrix between any two features is computed as

$$\mathbf{M}_{p,q} = \frac{1}{n_{p,q} - 1} \sum_{i \in I_{p,q}} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T, \tag{4.7}$$

where $I_{p,q}$ is the set of indices for training sample $\mathbf{x}_i$ belonging to class ω_p or ω_q, $n_{p,q}$ is the number of training samples, and $\overline{\mathbf{x}}$ is the mean feature vector over the $n_{p,q}$ training samples. Let $\mathbf{v}_{p,q}^k$, $k = 1, 2, \cdots, K$, are the top K ($K \leq P$) eigenvectors of matrix $\mathbf{M}_{p,q}$. We then assume that $\mathbf{A}_{p,q}$ is a linear combination of the top K eigenvectors:

$$\mathbf{A}_{p,q} = \sum_{k=1}^{K} \gamma_{p,q}^k \mathbf{v}_{p,q}^k \mathbf{v}_{p,q}^{k\ T}, \tag{4.8}$$

where $\gamma_{p,q}^k$, $k = 1, 2, \cdots, K$, are the non-negative weights for linear combination.

Then, with the above matrix $\mathbf{A}_{p,q}$, the overall log-likelihood for both the data pairs in $\mathcal{S}$ and $\mathcal{D}$ can be written as

$$\begin{aligned}
\mathcal{L}_g\left(\{\gamma_{p,q}^k\}_{k=1}^K, \mu_{p,q}\right) &= \log \Pr(+ \mid \mathcal{S}) + \log \Pr(- \mid \mathcal{D}) \\
&= -\sum_{(\mathbf{x}_m,\mathbf{x}_n)\in\mathcal{S}} \log\left(1 + \exp\left(d_{PM}^2(\mathbf{x}_m,\mathbf{x}_n) - \mu_{p,q}\right)\right) \\
&\quad -\sum_{(\mathbf{x}_m,\mathbf{x}_n)\in\mathcal{D}} \log\left(1 + \exp\left(-d_{PM}^2(\mathbf{x}_m,\mathbf{x}_n) + \mu_{p,q}\right)\right) \\
&= -\sum_{(\mathbf{x}_m,\mathbf{x}_n)\in\mathcal{S}} \log\left(1 + \exp\left(\sum_{k=1}^K \gamma_{p,q}^k \nu_{m,n}^k - \mu_{p,q}\right)\right) \\
&\quad -\sum_{(\mathbf{x}_m,\mathbf{x}_n)\in\mathcal{D}} \log\left(1 + \exp\left(-\sum_{k=1}^K \gamma_{p,q}^k \nu_{m,n}^k + \mu_{p,q}\right)\right),
\end{aligned} \tag{4.9}$$

with $\nu_{m,n}^k = (\mathbf{x}_m - \mathbf{x}_n)^T \mathbf{v}_{p,q}^k \mathbf{v}_{p,q}^{k\ T} (\mathbf{x}_m - \mathbf{x}_n)$.

With the maximum likelihood estimation, the PM distance metric learning can be formulated as the following optimization problem

$$\begin{aligned}
\max_{\{\gamma_{p,q}^k\}_{k=1}^K, \mu_{p,q}} \quad & \mathcal{L}_g\left(\{\gamma_{p,q}^k\}_{k=1}^K, \mu_{p,q}\right) \\
\text{s.t.} \quad & \gamma_{p,q}^k \geq 0,\ k = 1, \cdots, K,\ \text{and}\ \mu_{p,q} \geq 0,
\end{aligned} \tag{4.10}$$

which is similar to the optimization problem for the PW distance metric learning, and can be solved using the same optimization method.

4.3 Fusion of PkNN classifiers in the framework of belief functions

With the proposed pairwise distance metric concerning class pair $\Omega_{p,q}$, a pairwise kNN (PkNN) classifier can be designed to separate the two classes Ω_p and Ω_q based on the training subset $\mathcal{T}_p \cup \mathcal{T}_q$. For an M-class classification problem, $M(M-1)/2$ PkNN classifiers $\mathcal{C}_{p,q}$ $(1 \leq p < q \leq M)$ can be designed and the final classification result can be obtained by combining the outputs of these PkNN classifiers. A popular method of combining the outputs of pairwise classifiers is the voting rule [39], where each classifier gives a vote for the predicted class and the class with the largest number of votes is predicted. However, a classifier $\mathcal{C}_{p,q}$ is trained to distinguish only between classes Ω_p and Ω_q, thus its vote for a query pattern from a different class should be taken with care. In this section, we aim to overcome this difficulty by modeling the uncertainty related to the outputs of PkNN classifiers in the framework of belief functions. Figure 4.2 shows the evidential fusion scheme of the PkNN classifiers in the framework of belief functions.

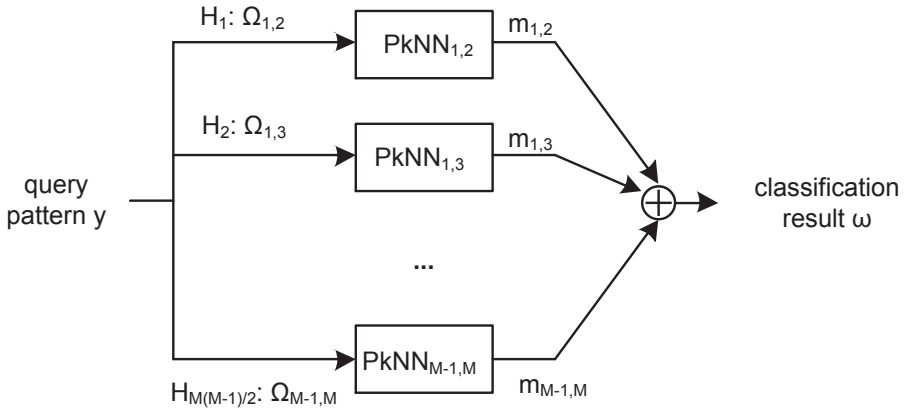


Figure 4.2: Fusion scheme of the PkNN classifiers in the framework of belief functions

Our aim is to use the belief function theory to model the uncertainty inherent in the pairwise classification. Now, with a set of PkNN classifiers $\mathcal{C}_{p,q}$ $(1 \leq p < q \leq M)$, we first study the representation of their outputs in terms of belief functions.

For the output of each PkNN classifier $\mathcal{C}_{p,q}$, there are two types of uncertainty. The first one is related to the fact that the real class label of query pattern $\mathbf{y}$ may actually not belong to class pair $\Omega_{p,q}$ (called *outer-pair uncertainty*). The second one is that even the real class label of query pattern $\mathbf{y}$ belongs to class pair $\Omega_{p,q}$, affected by the noise of the training patterns, the result of the classifier is not always accurate (called *inner-pair uncertainty*). Therefore, the frame of discernment should be set as $\Omega = \{\omega_1, \cdots, \omega_M\}$, which can characterize both kinds of uncertainty.

For the PkNN classifier $\mathcal{C}_{p,q}$, suppose $\mathbf{x}_j$ is one of the k nearest neighbors of query pattern $\mathbf{y}$ in the training subset $\mathcal{T}_p \cup \mathcal{T}_q$, and its class label is $\omega_i \in \Omega_{p,q}$. It can be seen as a piece of evidence that supports the query pattern $\mathbf{y}$ belongs to ω_i. However, considering the *outer-pair uncertainty*, this piece of evidence is conditioned on hypothesis $\omega_i \in \Omega_{p,q}$:

$$m^{\Omega}[\Omega_{p,q}](\{\omega_i\} \mid \mathbf{x}_j) = 1. \tag{4.11}$$

Further, due to the *inner-pair uncertainty*, this piece of evidence does not by itself provide 100% reliability. In the formalism of belief functions, this can be expressed by saying that only some part of the belief is committed to ω_i

$$\begin{cases} m^{\Omega}[\Omega_{p,q}](\{\omega_i\} \mid \mathbf{x}_j) &= \alpha_j \\ m^{\Omega}[\Omega_{p,q}](\{\Omega_{p,q}\} \mid \mathbf{x}_j) &= 1 - \alpha_j, \end{cases} \tag{4.12}$$

where $\alpha_j \in [0, 1]$ is the probability that sample $\mathbf{x}_j$ and query pattern $\mathbf{y}$ share the same class, and can be determined using the logistic regression model in Eq. (4.2) as

$$\alpha_j = \frac{1}{1 + \exp\left(d_{PW}^2(\mathbf{x}_j, \mathbf{y}) - \mu_{p,q}\right)}. \tag{4.13}$$

For the PkNN classifier $\mathcal{C}_{p,q}$, based on the k nearest neighbors of query pattern $\mathbf{y}$, we can calculate all the corresponding k mass functions in the above way. As the items of evidence from different neighbors are independent, the k mass functions are combined using Dempster's rule defined by Eq. (1.10) to form a resulting mass function synthesizing the overall conditional belief regarding the label of $\mathbf{y}$ as

$$m^{\Omega}[\Omega_{p,q}] = m^{\Omega}[\Omega_{p,q}](\cdot \mid \mathbf{x}_{i_1}) \oplus m^{\Omega}[\Omega_{p,q}](\cdot \mid \mathbf{x}_{i_2}) \oplus \cdots \oplus m^{\Omega}[\Omega_{p,q}](\cdot \mid \mathbf{x}_{i_k}), \tag{4.14}$$

where $i_1, i_2, \cdots, i_k$ are the indices of the k nearest neighbors of $\mathbf{y}$.

In a similar way, based on the outputs of the $M(M-1)/2$ PkNN classifiers $\mathcal{C}_{p,q}$ $(1 \leq p < q \leq M)$, we can calculate all the corresponding $M(M-1)/2$ conditional mass functions. In order to combine these conditional mass functions in a uniform framework, the conditional mass function constructed as Eq. (4.14) should be deconditioned using Eq. (1.23) as

$$\begin{cases} m^{\Omega}_{p,q}(\overline{\{\omega_q\}}) & = m^{\Omega}[\Omega_{p,q}](\{\omega_p\}) \\ m^{\Omega}_{p,q}(\overline{\{\omega_p\}}) & = m^{\Omega}[\Omega_{p,q}](\{\omega_q\}) \\ m^{\Omega}_{p,q}(\Omega) & = m^{\Omega}[\Omega_{p,q}](\Omega_{p,q}). \end{cases} \tag{4.15}$$

where $\overline{\{\omega_p\}}$ and $\overline{\{\omega_q\}}$ denote the complement of set $\{\omega_p\}$ and $\{\omega_q\}$ with respect to set Ω, respectively.

Because the mass and plausibility functions are in one-to-one correspondence, we can compute the plausibility function $Pl_{p,q}$ from the above constructed and deconditioned mass function $m^{\Omega}_{p,q}$ using Eq. (1.3) as

$$Pl_{p,q}(\{\omega_i\}) = \begin{cases} 1 - m^{\Omega}[\Omega_{p,q}](\{\omega_q\}), & \text{if } i = p \\ 1 - m^{\Omega}[\Omega_{p,q}](\{\omega_p\}), & \text{if } i = q \\ 1, & \text{otherwise .} \end{cases} \tag{4.16}$$

In order to decrease the computation complexity, instead of combining the $M(M-1)/2$ mass functions $m^{\Omega}_{p,q}$ $(1 \leq p < q \leq M)$ using Dempster's rule of combination, we can compute the combined plausibility function Pl directly using Eq. (1.25) to make the decision as follows

$$Pl(\{\omega_i\}) \propto Pl'(\{\omega_i\}) = \prod_{1 \leq p < q \leq M} Pl_{p,q}(\{\omega_i\}), \forall \omega_i \in \Omega. \tag{4.17}$$

Note that the combined plausibility function Pl is proportional to Pl', so the maximum plausibility rule can be used for Pl' equivalently to make a decision. The class label of query pattern $\mathbf{y}$ is assigned to the class with maximum plausibility.

Remark 4.2. *As can be seen from above, the fusion process of PkNN classifiers is quite time-efficient. Therefore, when classifying a query pattern, the time is mainly consumed in the classification process of multiple PkNN classifiers. Even though the number of PkNN*

classifiers is of M^2 order (with $M(M-1)/2$ classifiers), each classifier only uses the training samples from the corresponding classes (about $2N/M$ samples averagely). Hence the total number of the computed samples is about $N(M-1)$, which is just $M-1$ times larger than the original kNN classifier. For most classification problems, such as the benchmark data sets studied in next section, the number of considered classes is not very large, so the computation cost of the proposed method is not a big problem.

4.4 Experiments

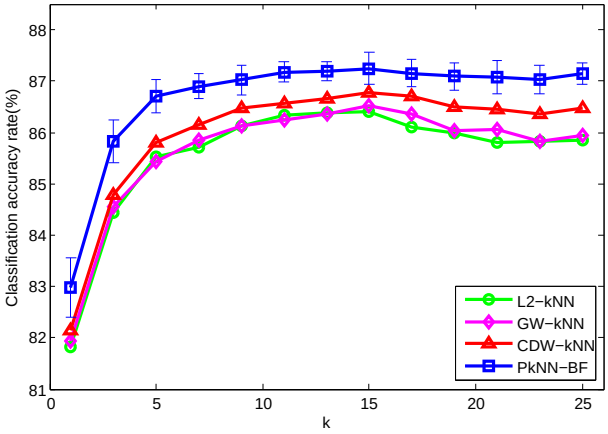
The performance of the proposed kNN classification method based on the pairwise distance metric and the belief function theory (denoted as PkNN-BF) was assessed by two different types of experiments. In the first experiment, a synthetic data set was used to show the behavior of the proposed method in a controlled setting. In the second one, several real data sets from the well-known UCI Repository of Machine Learning Databases [72] were considered, with the aim to show that the proposed technique is adequate for a variety of real tasks involving different data conditions: large vs. small size, high vs. low dimension, etc.

4.4.1 Synthetic data test

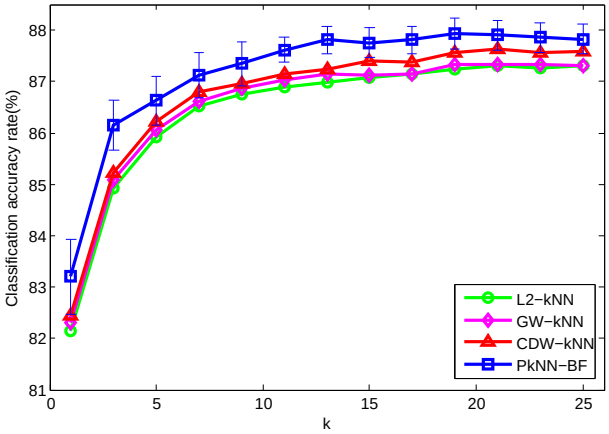
A two-dimensional three-class classification problem was used to compare our method with the-state-of-art methods reviewed in Section 2.3.3, including the original kNN classifier based on L2 distance metric (L2-kNN) [38], the kNN classifier based on GW distance metric (GW-kNN) [124] and the kNN classifier based on CDW distance metric (CDW-kNN) [79]. The following class-conditional normal distributions were assumed.

$$\begin{array}{lll} \mu_A = (6,6)^T, & \mu_B = (14,6)^T, & \mu_C = (14,14)^T, \\ \Sigma_A = 3\mathbf{I}, & \Sigma_B = 3\mathbf{I}, & \Sigma_C = 3\mathbf{I}. \end{array}$$

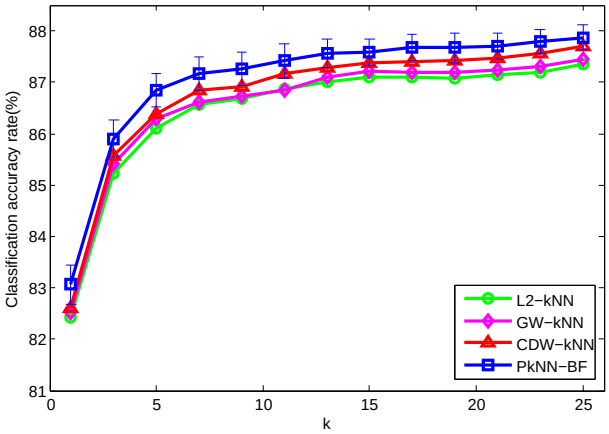
Training sets of 60, 120 and 240 samples were generated using equal prior probabilities. A test set of 3000 samples was used for classification accuracy estimation. For each case, 30 trials were performed with independent training sets. The average classification accuracy and the corresponding 95% confidence interval were calculated. For the proposed PkNN-BF method, as the features are independent from each other in this study, we used the PW distance metric. For all of the considered methods, values of k ranging from 1 to 25 have been investigated. Figure 4.3 shows the classification accuracy of the considered methods with different training set sizes. As can be seen from the results, the GW-kNN classifier shows similar performance as compared to the original L2-kNN classifier, because the learned GW distance metric has almost the same weights for the two involved features. The CDW-kNN classifier, which is based on the CDW distance metric, is just slightly



(a) N=60



(b) N=120



(c) N=240

Figure 4.3: Classification accuracy rate (in %) for synthetic data with different training set sizes (L2-kNN: kNN based on L2 distance metric, GW-kNN: kNN based on GW distance metric, CDW-kNN: kNN based on CDW distance metric, PkNN-BF: kNN based on pairwise distance metric and the belief function theory)

better than the original L2-kNN classifier. The proposed PkNN-BF classifier produces the highest classification accuracy for all of the three cases. The reason is that, for each PkNN sub-classifier, the pairwise distance metric characterizes more local specificities in feature space, and further in the combination process, the output uncertainty of those PkNN sub-classifiers is well addressed. In addition, the performance improvement is more significant for small training set, in which case the distance metric plays a more important role in determining the performance of the kNN-based classifiers.

To better illustrate the superiority of the proposed PkNN-BF classifier, the classification results of one test sample $\mathbf{y}$ (with real label Class B) for different methods using 60 training samples are shown in Figure 4.4. To visualize the results, for the GW-kNN classifier and the three pairwise sub-classifiers (i.e., PkNN$_{A,B}$, PkNN$_{A,C}$ and PkNN$_{B,C}$), each original point $\mathbf{x}$ is replaced by $\mathbf{Ax}$, where $\mathbf{A}$ is a diagonal matrix filled by the learned feature weights. After this procedure, the classification problem is transformed into applying the standard Euclidean metric to the rescaled data to find the nearest neighbors[1]. As can be seen in Figure 4.4(a), test sample $\mathbf{y}$ is quite close to the boundaries of the three classes, and in this small training set condition, it is quite difficult to make the right classification. The L2-kNN classifier just misclassifies this data point with the 1NN rule. For the GW-kNN classifier, as the learned global feature weights are almost equivalent ($\lambda_X = 0.2180$, $\lambda_Y = 0.2033$), the rescaled data distributions are quite similar with the original distributions. Accordingly, as can be seen in Figure 4.4(b), the GW-kNN classifier also misclassifies the test sample $\mathbf{y}$. For our proposed PkNN-BF classifier, three sub-classifiers with separately learned distance metrics are designed to classify the test sample $\mathbf{y}$. In classifying $\mathbf{y}$ between Class A and Class B, feature X is assigned larger weight ($\lambda^X_{A,B} = 0.2231$, $\lambda^Y_{A,B} = 0.0357$), whereas in classifying $\mathbf{y}$ between Class B and Class C, feature Y is assigned larger weight ($\lambda^X_{B,C} = 0.0265$, $\lambda^Y_{B,C} = 0.2156$). Thanks to this locally learned pairwise distance metric, as shown in Figure 4.4(c) and (e), both the two sub-classifiers PkNN$_{A,B}$ and PkNN$_{B,C}$ provide the correct classification result. The PkNN-BF classifier classifies test sample $\mathbf{y}$ by combining the results of the three PkNN sub-classifiers:

$$\begin{array}{lll} Pl_{A,B}(\{A\}) = 0.17, & Pl_{A,B}(\{B\}) = 1, & Pl_{A,B}(\{C\}) = 1; \\ Pl_{A,C}(\{A\}) = 1, & Pl_{A,C}(\{B\}) = 1, & Pl_{A,C}(\{C\}) = 0.42; \\ Pl_{B,C}(\{A\}) = 1, & Pl_{B,C}(\{B\}) = 1, & Pl_{B,C}(\{C\}) = 0.22. \end{array}$$

Then, after the fusion of multiple PkNN sub-classifiers in the framework of belief functions, we get the combined result:

$$Pl'(\{A\}) = 0.17,\ Pl'(\{B\}) = 1,\ Pl'(\{C\}) = 0.09.$$

Finally, based on the maximum plausibility rule, we get Class B as the final classification.

[1] As the CDW distance metric does not satisfy the symmetry property, it is impossible to rescale the data to apply the standard Euclidean metric, so we did not visualize the result of the CDW-kNN in this illustration.

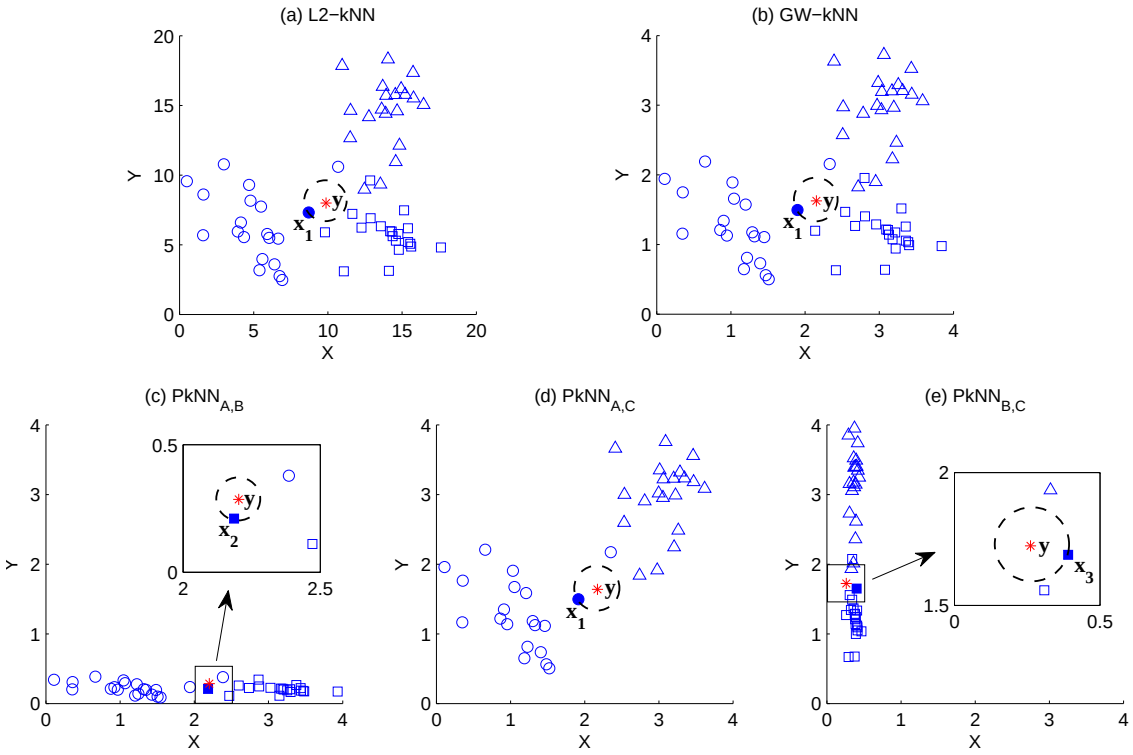


Figure 4.4: Classification results of test sample $\mathbf{y}$ for different methods with 60 training samples (with '$\circ$' for class A, '$\square$' for class B and '$\triangle$' for class C, respectively)

4.4.2 Real data test

In this second experiment, ten well-known real data sets from the UCI repository were used to evaluate the performance of the PkNN-BF classifier. The main characteristics of the data sets are summarized in Table 4.1. In order to evaluate the effectiveness of the combination process using the belief functions, apart from the above compared methods, we also considered the method of combining the PkNN sub-classifiers using the voting rule (denoted as PkNN-VOTE). As for these real data sets, the feature correlations are unknown, the PM distance metric has been used for the PkNN sub-classifiers, and the extension of GW-kNN in Mahalanobis distance (denoted as GM-kNN) [124] was instead for comparison. For all these kNN-based methods, the results for the optimal value of k ranging from 1 to 25, observed in each method were reported.

The classification results of the ten benchmark data sets are shown in Table 4.2. The significance of the differences between results is evaluated using a *Mc Nemar* test [34] at a level of significance of $\alpha = 0.05$. For each data set, the best classification accuracy is underlined, and those that are significantly improved than the baseline L2-kNN method are printed in bold. As can be seen from these results, the PkNN-BF method, presented in this chapter, outperforms the L2-kNN, GM-kNN and CDW-kNN methods for most of the

Table 4.1: Statistics of the benchmark data sets used in the experiment

Data set	# Instances	# Features	# Classes	# Training	# Testing
Balance	625	4	3	500	125
Ecoli	336	7	8	200	136
Glass	214	9	6	139	75
Satimage	6,435	36	6	4,435	2,000
Segment	2,310	19	7	1,400	910
Vehicle	846	18	4	646	200
Vertebral	310	6	3	150	160
Waveform	5,000	21	3	3,500	1,500
Wine	178	13	3	75	103
Yeast	1,484	8	10	1,000	484

data sets. Additionally, for *Ecoli*, *Glass*, *Satimage*, *Segment*, *Vehicle*, *Waveform* and *Wine* data sets, the improvements are statistically significant than the baseline L2-kNN method, because the local distance metric plays more crucial role in determining the kNN-based classification performance for these small-size and high-dimension cases. In addition, the PkNN-BF method always performs better than PkNN-VOTE, especially for those data sets with small number of classes, because the voting rule will take great adventure when the total number of votes ($M(M-1)/2$, with M be the number of classes) is small.

Table 4.2: Classification accuracy rate (in %) of our proposed method compared with other kNN-based methods for real data

Data set	L2-kNN	GM-kNN	CDW-kNN	PkNN-VOTE	PkNN-BF
Balance	88.40	89.60	91.20[a]	89.60	90.40
Ecoli	84.56	84.56	85.29	**87.50**[b]	**88.24**
Glass	69.33	70.67	70.67	72.00	**74.67**
Satimage	89.45	**91.55**	89.30	**91.95**	**92.55**
Segment	95.05	96.15	94.51	**96.70**	**96.92**
Vehicle	69.50	70.50	69.50	72.00	**74.50**
Vertebral	83.88	83.25	84.50	83.88	85.12
Waveform	80.67	81.07	**84.07**	**84.53**	**85.93**
Wine	77.67	78.64	**89.32**	**88.35**	**93.20**
Yeast	56.82	55.58	55.79	57.02	57.23

[a]The results underlined correspond to the best accuracy.

[b]The results typeset in boldface are significantly better than the baseline L2-kNN method at level $\alpha = 0.05$.

4.5 Conclusion

In order to improve the performance of the kNN-based classifier in incomplete data set situations, a new distance metric called pairwise distance metric, has been proposed in this chapter. Compared with the existing distance metrics, the pairwise distance metric provides greater flexibility to design the feature weights so that the local specificities in feature space can be well characterized. A parameter optimization procedure was designed to learn the pairwise distance metric from the training data set. Based on the pairwise distance metric, a PkNN-BF classifier was developed, which combines the outputs of PkNN classifiers in the framework of belief functions. From the results reported in the last section, we can conclude that the proposed method achieved a uniformly good performance when applied to a variety of classification tasks, including those with high dimension and small sample size, in which cases the training data set is not rich enough to well characterize the real class-conditional probability distributions.

Part III

Rule-based classification

This part focuses on classification of uncertain data using rule-based approaches.

Chapter 5 focuses on improving the performance of the rule-based classification system in complex applications. We extend the traditional rule-based classification system in the framework of belief functions and develop a belief rule-based classification system to address uncertain information in complex classification problems.

Chapter 6 concerns the classification problems based on partially available training data and expert knowledge. A hybrid belief rule-based classification system is developed to make use of these two types of information jointly for classification.

Belief rule-based classification system

Among the computational intelligence techniques employed to solve classification problems, the fuzzy rule-based classification system (FRBCS) is a popular tool capable of building a linguistic model interpretable to users. However, it may face lack of accuracy in some complex applications, because the inflexibility of the concept of the linguistic variable imposes hard restrictions on the fuzzy rule structure. In this chapter, we extend the FRBCS with a belief rule structure and develop a belief rule-based classification system (BRBCS) to address the uncertain information in complex classification problems. The two components of the proposed BRBCS, i.e., the belief rule base and the belief reasoning method, are designed specifically by taking into account the pattern noise that exists in many real-world data sets.

In this chapter, we first give an introduction about the background and motivations in Section 5.1. Then, the belief rule-based classification system is developed within the framework of belief functions in Section 5.2. Three experiments are then performed in Section 5.3 to evaluate the accuracy, robustness, and time complexity of the proposed method. Finally, Section 5.4 concludes this chapter.

5.1 Introduction

The fuzzy rule-based classification system (FRBCS), first developed by Chi et al. [17], has become a popular framework for classifier design due to its capability of building a linguistic model interpretable to users. However, the FRBCS may have low accuracy when dealing with some complex applications, due to the inflexibility of the concept of the linguistic variable, which imposes hard restrictions on the fuzzy rule structure [5]. Besides, the fuzzy rule structure is also not robust to pattern noise, which hinders its applications in harsh working conditions. As reviewed in Section 2.4.2, plenty of work has been done in the past two decades in order to improve the accuracy and robustness of the FRBCS.

In fact, different types of uncertainty, such as fuzziness, imprecision and incompleteness,

may coexist in real-world complex systems. The FRBCS, which is based on fuzzy set theory [132], cannot effectively address imprecise or incomplete information in the modeling and reasoning processes. The theory of belief functions, proposed and developed by Dempster [24] and Shafer [96], has become one of the most powerful frameworks for uncertain modeling and reasoning. As the fuzzy set theory is well suited to dealing with fuzziness, and the belief function theory provides an ideal framework for handling imprecision and incompleteness, many researchers have investigated the relationship between fuzzy set theory and belief function theory and suggested different methods of integrating them [12, 15, 65, 127, 128]. Among these methods, Yang et al. [128] extended the fuzzy rule in the framework of belief functions and proposed a new knowledge representation scheme in a belief rule structure, which is capable of capturing fuzzy, imprecise, and incomplete causal relationships. The belief rule structure has been successfully applied in clinical risk assessment [57], inventory control [63], fault diagnosis [135], and new product development [106, 129].

In this chapter, we aim to extend the fuzzy rule in FRBCS with the belief rule structure developed in [128] for classification applications. Compared with the fuzzy rule, the consequence part of the belief rule is in a belief distribution form, which is more informative and can characterize the uncertain information (i.e., fuzziness, imprecision, and incompleteness) existing in the training set. In addition, feature weights are introduced in the belief rule to characterize the different degrees of importance of features to the consequence. Therefore, the belief rule is more suitable for modeling those complex classification problems with high uncertainty. Based on the belief rule structure, a belief rule-based classification system (BRBCS) is developed as an extension of the FRBCS in the framework of belief functions. In the proposed BRBCS, a data-driven belief rule base (BRB) generation method is developed to establish the uncertain association between the feature space and the class space. This BRB generation method enables the automatic generation of belief rules from the training data without the requirement of a priori expert knowledge. Then, to classify a query pattern based on the BRB, a belief reasoning method (BRM) is developed based on the belief function theory. This BRM can well address the uncertainty existing in the consequences of activated belief rules for a query pattern.

To handle the pattern noise commonly existing in many real-world data sets, two techniques are developed in the BRB generation and BRM design processes. First, the consequence part of each belief rule in BRB is generated by fusing the information coming from all of the training samples assigned to the corresponding antecedent fuzzy region. In this way, the adverse effects of the noisy training samples on the consequence of the belief rule can be reduced to some extent. Furthermore, in BRM, the final consequent class of a query pattern is obtained by combining the consequence parts of all of the belief rules activated by the query pattern. Thus, even if some unreliable belief rules are generated in noisy conditions, this procedure can further reduce the risk of misclassification.

5.2 Belief rule-based classification system

Considering the advantages of belief functions for representing and reasoning with uncertain information, in this section we extend the classical FRBCS in the framework of belief functions and develop a belief rule-based classification system (BRBCS). As shown in Figure 5.1, the proposed BRBCS is composed of two components: the belief rule base, which establishes an association between the feature space and the class space, and the belief reasoning method which provides a mechanism to classify a query pattern based on the constructed rule base. In Section 5.2.1, we first describe the belief rule structure for classification applications, which extends the traditional fuzzy rule structure in the framework of belief functions. Based on the belief rule structure, we learn the belief rule base from the training data in Section 5.2.2, and then the belief reasoning method is developed in Section 5.2.3.

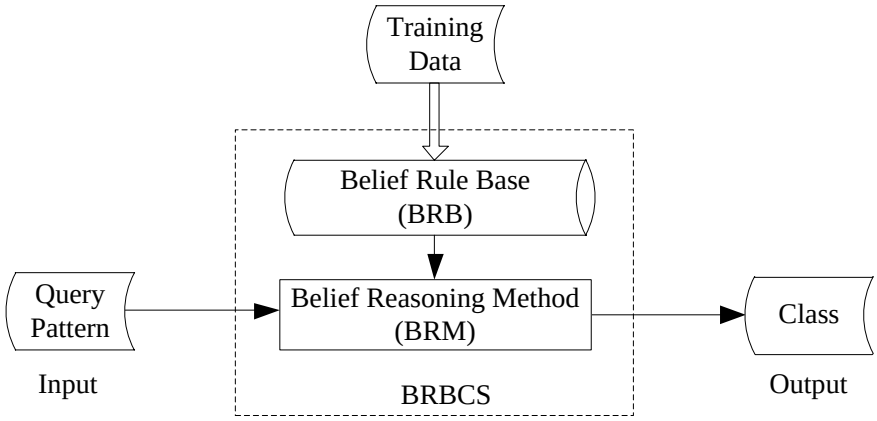


Figure 5.1: Belief rule-based classification system

5.2.1 Belief rule structure

The fuzzy rule structure expressed in Eq. (2.10) is relatively simple in that it does not consider the distribution of consequence and the relative importance of each feature. To take the above aspects into consideration, two concepts are introduced [128]:

- *Belief degrees of consequence.* For a complex classification problem, it is likely that the consequence of a rule may take a few values with different belief degrees. Suppose the consequence may have M different classes, $\omega_1, \omega_2, \cdots, \omega_M$, and the corresponding belief degrees are represented by $\beta_i (i = 1, 2, \cdots, M)$, then the consequence with a belief structure can be represented by $\{(\omega_1, \beta_1), (\omega_2, \beta_2), \cdots, (\omega_M, \beta_M)\}$.
- *Feature weights.* In real-world classification problems, different features may behave distinctly in determining the consequent class. Thus, there is a need to assign a weight to each feature to describe such degrees of importance.

To take into account the belief degrees of consequence and the feature weights, the fuzzy rule structure expressed in Eq. (2.10) can be extended to the following belief rule structure for classification purposes:

$$\begin{aligned}&\text{Belief Rule } R^q:\\&\text{If } x_1 \text{ is } A_1^q \text{ and } \cdots \text{ and } x_P \text{ is } A_P^q, \text{ then consequence is } \mathbf{C}^q = \{(\omega_1, \beta_1^q), \cdots, (\omega_M, \beta_M^q)\}\\&\text{with rule weight } \theta^q \text{ and feature weights } \delta_1, \cdots, \delta_P, \quad q = 1, 2, \cdots, Q,\end{aligned} \tag{5.1}$$

where β_m^q is the belief degree to which ω_m is believed to be the consequent class for the q-th belief rule. In the belief structure, the consequence may be incomplete, i.e., $\sum_{m=1}^{M} \beta_m^q < 1$, and the left belief $1 - \sum_{m=1}^{M} \beta_m^q$ denotes the degree of ignorance about the class label. The rule weight θ^q with $0 \leq \theta^q \leq 1$ characterizes the certainty grade of the belief rule R^q and the feature weights $\delta_1, \cdots, \delta_P$ with $0 \leq \delta_1, \cdots, \delta_P \leq 1$ describe the importance of different features in determining the consequent class.

Remark 5.1. *Compared with the fuzzy rule structure, the belief rule structure has some advantages for classification problems as follows. a) In the belief rule structure, the consequence is in a belief distribution form. On the one hand, with the distribution form, any difference in the antecedent part can be clearly reflected in the consequence, whereas in a traditional fuzzy rule, different antecedents may lead to the same consequence. On the other hand, by introducing belief functions, the belief structure makes the rule more appropriate to characterize the uncertain information. In generating each rule, only limited training samples are available, and each training sample only provides partial evidence about the consequence of this rule. Thus, the corresponding consequence of this rule should not be complete. The belief rule structure can well characterize this incompleteness, with the remained belief* $1 - \sum_{m=1}^{M} \beta_m^q$ *denoting the degree of ignorance about the class label induced by the limited training samples. b) With the introduction of feature weights, the importance of different features to the consequence can be well characterized, which is closer to reality. In summary, compared with the traditional fuzzy rule, the belief rule is more informative, more flexible and thus more suitable for modeling those complex classification problems.*

5.2.2 Belief rule base generation

To make a classification with BRBCS, the first step is to generate a BRB from the training set. The FRB generation method given by Chi et al. [17] is used as a base model in this section to develop the BRB generation method in the framework of belief functions. As displayed in Eq. (5.1), each belief rule is composed of four components, namely, the antecedent part, the belief degrees of consequence, the rule weight and the feature weights. Because the antecedent part in the belief rule is the same as that in the fuzzy rule, here we only focus on the generation of the latter three components, i.e., the belief degrees of consequence, the rule weight and the feature weights.

5.2.2.1 Generation of belief degrees of consequence

In this part, with the symbols defined in Section 5.2.1, we develop an algorithm to generate the belief degrees of consequence in BRB.

Similar to the generation of the consequent class in FRB in the first step, we also need to calculate the matching degree $\mu_{\mathbf{A}^q}(\mathbf{x}_i)$ of each training sample $\mathbf{x}_i$ with the antecedent part $\mathbf{A}^q$ using Eq. (2.11). In FRB, the consequent class is directly specified as the class label of the training sample having the greatest matching degree with the antecedent part $\mathbf{A}^q$. However, this procedure may entail great risk, especially when class noise exists in the training set. In BRB, we fuse the class information of all of the training samples assigned to the corresponding antecedent fuzzy region to get the consequence in a belief distribution form.

Denote as $\mathcal{T}^q$ the set of training samples assigned to the antecedent fuzzy region $\mathbf{A}^q$. From the view of belief functions, the class set $\Omega = \{\omega_1, \cdots, \omega_M\}$ can be regarded as the frame of discernment of the problem. For any training sample $\mathbf{x}_i \in \mathcal{T}^q$, the class label $\text{Class}(\mathbf{x}_i) = \omega_m$ can be regarded as a piece of evidence that increases the belief that the consequent class belongs to ω_m. However, this piece of evidence does not by itself provide 100% certainty. In the framework of belief functions, this can be expressed by saying that only some part of the belief (measured by the matching degree $\mu_{\mathbf{A}^q}(\mathbf{x}_i)$) is committed to ω_m. Because $\text{Class}(\mathbf{x}_i) = \omega_m$ does not point to any other particular class, the rest of the belief should be assigned to the frame of discernment Ω representing ignorance. Therefore, this item of evidence can be represented by a mass function m_i^q verifying:

$$\begin{cases} m_i^q(\{\omega_m\}) &= \mu_{\mathbf{A}^q}(\mathbf{x}_i) \\ m_i^q(\Omega) &= 1 - \mu_{\mathbf{A}^q}(\mathbf{x}_i) \\ m_i^q(A) &= 0, \quad \forall A \in 2^\Omega \setminus \{\Omega, \{\omega_m\}\}, \end{cases} \tag{5.2}$$

with $0 < \mu_{\mathbf{A}^q}(\mathbf{x}_i) \leq 1$.

For each $\mathbf{x}_i \in \mathcal{T}^q$, a mass function depending on both its class label and its matching degree with the antecedent part can therefore be defined. To obtain the consequence associated with the antecedent part $\mathbf{A}^q$ in a belief distribution form, these mass functions can be combined using Dempster's rule. As shown in Eq. (5.2), only two focal elements are involved in each mass function. Because of the particular structure of the mass function, the computational burden of Dempster's rule can be greatly reduced, and the analytical

formulas can be derived as

$$\begin{cases} m^q(\{\omega_m\}) &= \dfrac{1}{1-K^q}\left(1-\prod\limits_{\mathbf{x}_i\in\mathcal{T}_m^q}(1-\mu_{\mathbf{A}^q}(\mathbf{x}_i))\right)\prod\limits_{r\neq m}\prod\limits_{\mathbf{x}_i\in\mathcal{T}_r^q}(1-\mu_{\mathbf{A}^q}(\mathbf{x}_i)), \\ & \quad m=1,2,\cdots,M, \\ m^q(\Omega) &= \dfrac{1}{1-K^q}\prod\limits_{r=1}^{M}\prod\limits_{\mathbf{x}_i\in\mathcal{T}_r^q}(1-\mu_{\mathbf{A}^q}(\mathbf{x}_i)), \end{cases} \tag{5.3}$$

where $\mathcal{T}_m^q$ is a subset of $\mathcal{T}^q$, corresponding to those training samples belong to class ω_m, and K^q is the total conflicting belief mass

$$K^q = 1+(M-1)\prod_{r=1}^{M}\prod_{\mathbf{x}_i\in\mathcal{T}_r^q}(1-\mu_{\mathbf{A}^q}(\mathbf{x}_i)) - \sum_{m=1}^{M}\prod_{r\neq m}\prod_{\mathbf{x}_i\in\mathcal{T}_r^q}(1-\mu_{\mathbf{A}^q}(\mathbf{x}_i)). \tag{5.4}$$

Therefore, the belief degrees of consequence of rule R^q can be obtained as

$$\begin{cases} \beta_m^q &= m^q(\{\omega_m\}), \quad m=1,2,\cdots,M, \\ \beta_\Omega^q &= m^q(\Omega), \end{cases} \tag{5.5}$$

where β_Ω^q is the belief degree unassigned to any individual class.

Remark 5.2. *In the classical FRBCS reviewed in Section 2.4.1, the consequence of each rule is only determined by the class label of the training sample having the greatest matching degree with the antecedent part, whereas the consequence in the belief rule fuses information that comes from all of the training samples assigned to the corresponding antecedent fuzzy region. Thus, it can effectively reduce the adverse effects of some noisy training samples. The method to generate the consequence is similar to some data-cleaning approaches [6,22]. The difference is that the data-cleaning approaches remove the unreliable training samples, whereas our method retains all training samples and generates the consequence in a belief distribution form, which can be considered as soft labels. Compared with the data cleaning approaches, the belief distribution form maintains more information from the training samples and can be further combined with the consequences of other rules in later processing.*

Remark 5.3. *The idea to generate the belief degrees of consequence in this chapter is inspired by the EkNN classification method developed by Denœux [26], in which each of the k nearest neighbors of the query pattern is considered as an item of evidence that supports certain hypotheses regarding the class membership of that training sample. The corresponding relations of the two methods are as follows: a) the training samples assigned to the corresponding antecedent fuzzy region correspond to the k nearest neighbors of the query pattern in EkNN, and b) the validity of each training sample is measured by the matching degree with the antecedent part, and in EkNN, that is measured by the distance from the query pattern. With the above relationship, it can be further deduced that the consequence generation method in FRB reviewed in Section 2.4.1 has a similar idea to the*

voting kNN classification method [38]. As illustrated in [26], the EkNN classifier can obtain much better performance than the voting kNN classifier, especially in noisy conditions. Thus, it is expected that the belief distribution form of consequence in the BRB can handle the class noise more effectively than the single class form of consequence in the FRB.

5.2.2.2 Generation of rule weights

In the area of data mining, two measures called *confidence* and *support* have often been used for evaluating association rules [2]. Our belief rule R^q in Eq. (5.1) can be viewed as a type of association rule of the form $\mathbf{A}^q \Rightarrow \mathbf{C}^q$. The main difference from the standard formulation of the association rule is that in our belief IF-THEN rule, the input variable is in fuzzy form and the output variable is in belief distribution form. In this part, we will draw the rule weight θ^q from the concepts of confidence and support.

The confidence is defined as a measure of the validity of one association rule [2]. For our belief IF-THEN rule, the consequence part $\mathbf{C}^q$ is obtained by combining the items of evidence coming from all of the training samples assigned to the antecedent fuzzy region $\mathbf{A}^q$. It is believed that if the items of evidence involved are in conflict with each other (e.g., if the items of evidence assign different classes with the highest belief), then the consequence has low validity. In the framework of belief functions, several models are proposed to measure the conflict among different items of evidence [66, 102]. The conflict factor $\sum_{B\cap C=\emptyset} m_1(B)m_2(C)$ derived in Dempster's rule is employed here for its simplicity and convenience. The confidence of the belief rule R^q is hence defined as

$$c(R^q) = 1 - \overline{K^q}, \tag{5.6}$$

with the average conflict factor $0 \leq \overline{K^q} \leq 1$ calculated by

$$\overline{K^q} = \begin{cases} 0, & \text{if } |\mathcal{T}^q| = 1, \\ \dfrac{1}{|\mathcal{T}^q|(|\mathcal{T}^q|-1)} \displaystyle\sum_{\mathbf{x}_i,\mathbf{x}_j\in\mathcal{T}^q;i<j} \sum_{B\cap C=\emptyset} m_i^q(B)m_j^q(C), & \text{otherwise.} \end{cases} \tag{5.7}$$

where $|\mathcal{T}^q|$ is the number of training samples assigned to the fuzzy region $\mathbf{A}^q$.

On the other hand, as described in [2], the support indicates the grade of the coverage by one association rule. For our belief IF-THEN rule, N training samples are available for rule generation, whereas only those assigned to the corresponding antecedent fuzzy region are used to generate the consequence. Therefore, the support of the belief rule R^q is defined as

$$s(R^q) = \frac{|\mathcal{T}^q|}{N}. \tag{5.8}$$

As defined above, the confidence and support characterize the weight of the belief rule in two distinct aspects and should therefore be considered jointly. On the one hand, if the

belief rule R^q has high confidence but low support (e.g., if only one training sample is assigned to the antecedent fuzzy region $\mathbf{A}^q$), the belief rule weight should be decreased, as the consequence may be easily affected by the class noise. On the other hand, if the belief rule R^q has high support but low confidence (e.g., if a large number of training samples are contained in $\mathcal{T}^q$ but with great divergence in the class label), the belief rule weight should also be decreased, considering the great conflicts. The product of the confidence $c(R^q)$ and the support $s(R^q)$ is used to characterize the weight of the belief rule R^q as

$$\theta^q \propto c(R^q)s(R^q). \tag{5.9}$$

Following a normalization process, we obtain the weights of all of the belief rules as

$$\theta^q = \frac{c(R^q)s(R^q)}{\max\limits_q\{c(R^q)s(R^q), q = 1, \cdots, Q\}}, \quad q = 1, 2, \cdots, Q. \tag{5.10}$$

5.2.2.3 Generation of feature weights

In the belief rule displayed as Eq. (5.1), the feature weights reflect the relative importance of the antecedent features with respect to their influence on the consequence. An antecedent feature with a higher weight is more influential on the consequence. Therefore, to determine a feature weight is to find a way to measure the relative intensity of the influence that this antecedent feature imposes on the consequence in comparison with others. In this part, such a measurement is quantified by a so-called *correlation factor* between each feature and the consequence.

Suppose n_p fuzzy partitions $\{A_{p,1}, A_{p,2}, \cdots, A_{p,n_p}\}$ are established for feature A_p. Now, we will derive the weight of feature A_p by correlation analysis with the corresponding consequence. Specifically, we use the relationship between the changes of different fuzzy partitions that A_p takes and the changes of the consequence to determine the correlation between A_p and the consequence.

As Q belief rules with different antecedent parts are available in the BRB, in the first place, for feature A_p, according to its n_p fuzzy partitions $\{A_{p,1}, A_{p,2}, \cdots, A_{p,n_p}\}$, we divide the BRB into n_p sub-BRBs B_k, $k = 1, 2, \cdots, n_p$, with each sub-BRB B_k containing all of the belief rules using the fuzzy partition $A_{p,k}$ for feature A_p:

$$B_k = \left\{R^q \mid A_p^q = A_{p,k}, q = 1, 2, \cdots, Q\right\}, \quad k = 1, 2, \cdots, n_p. \tag{5.11}$$

Then, for each sub-BRB B_k, the consequence parts of all of the contained belief rules are combined to obtain the integrated consequence m_k with the weighted averaging operation:

$$\begin{cases} m_k(\{\omega_m\}) &= \dfrac{1}{\sum\limits_{R^q \in B_k} \theta^q} \sum\limits_{R^q \in B_k} \theta^q \beta_m^q, \quad m = 1, 2, \cdots, M, \\ m_k(\{\Omega\}) &= \dfrac{1}{\sum\limits_{R^q \in B_k} \theta^q} \sum\limits_{R^q \in B_k} \theta^q \beta_\Omega^q, \end{cases} \tag{5.12}$$

where β_m^q, $m = 1, 2, \cdots, M$, and β_Ω^q are the belief degrees of consequence of rule R^q generated in Section 5.2.2.1, and θ^q is the rule weight of R^q generated in Section 5.2.2.2.

Thus, when A_p changes its fuzzy partition from $A_{p,k}$ to $A_{p,k+1}$, $k = 1, 2, \cdots, n_p - 1$, the change of the consequence is

$$\Delta C_{p,k} = d_J(m_k, m_{k+1}), \tag{5.13}$$

where d_J is Jousselme's distance, as defined by Eq. (1.8).

Then, the average change of the consequence for k changing from 1 to $n_p - 1$ is obtained as

$$\Delta C_p = \frac{\sum_{k=1}^{n_p-1} \Delta C_{p,k}}{n_p - 1}. \tag{5.14}$$

In this chapter, we define ΔC_p as the correlation factor (CF) between feature A_p and the consequence, i.e.,

$$CF_p = \Delta C_p. \tag{5.15}$$

In a similar way, we obtain the correlation factors $CF_p, p = 1, 2, \cdots, P$ for all features. Further, δ_p, the weight of feature A_p, can be generated from the normalized CF_p as follows

$$\delta_p = \frac{CF_p}{\max\limits_p \{CF_p, p = 1, 2, \cdots, P\}}, \quad p = 1, 2, \cdots, P. \tag{5.16}$$

5.2.3 Belief reasoning method

As reviewed in Section 2.4.1, in FRBCS, the single winner FRM is used to classify a new query pattern. However, when excessive noise exists in the training set, this method may have a great risk of misclassification. In this section, we will fuse the consequences of all of the rules activated by the query pattern in the framework of belief functions to get a more robust classification. The main idea is firstly calculating the association degrees of the query pattern with the consequences of the activated belief rules and then combining these consequences with respect to their reliability (characterized by the association degrees).

5.2.3.1 Association degree with the consequence of a belief rule

Denote as $\mathbf{y} = (y_1, y_2, \cdots, y_P)$ a query pattern to be classified. In the first place, the matching degree of the query pattern with the antecedent part of each rule is calculated. As the feature weights are complemented in the belief rule, we use the following simple weighted multiplicative aggregation function to calculate the matching degree

$$\mu_{\mathbf{A}^q}(\mathbf{y}) = \sqrt[P]{\prod_{p=1}^{P} \left[\mu_{A_p^q}(y_p)\right]^{\delta_p}}, \tag{5.17}$$

where $\mu_{A_p^q}$ is the membership function of the antecedent fuzzy set A_p^q, and δ_p is the weight of the p-th feature given in Eq. (5.16).

Remark 5.4. *In Eq. (5.17), the contribution of a feature towards the matching degree is positively related to the weight of the feature. A more important feature plays a greater role in determining the matching degree. Particularly, if* $\delta_p = 0$, *then* $\left[\mu_{A_p^q}(y_p)\right]^{\delta_p} = 1$, *which shows that a feature with zero importance does not have any impact on the matching degree; if* $\delta_p = 1$, *then* $\left[\mu_{A_p^q}(y_p)\right]^{\delta_p} = \mu_{A_p^q}(y_p)$, *which shows that the most important feature has the largest impact on the matching degree.*

Let $\mathcal{S}$ be the set of Q constructed belief rules in the BRB. Denote as $\mathcal{S}' \subseteq \mathcal{S}$ the set of belief rules activated by query pattern $\mathbf{y}$:

$$\mathcal{S}' = \{R^q \mid \mu_{\mathbf{A}^q}(\mathbf{y}) \neq 0, q = 1, 2, \cdots, Q\}. \tag{5.18}$$

The association degree of query pattern $\mathbf{y}$ with the consequence of one activated belief rule $R^q \in \mathcal{S}'$ is determined by two factors, the matching degree and the rule weight. The matching degree reflects the similarity between the query pattern and the antecedent part of the belief rule, and the rule weight characterizes the reliability of the belief rule. Thus, the association degree is defined as

$$\alpha^q = \mu_{\mathbf{A}^q}(\mathbf{y})\theta^q, \quad \text{for } R^q \in \mathcal{S}'. \tag{5.19}$$

Remark 5.5. *As a result of limitation due to the number of training samples, in some applications, there may be no rule activated by query pattern* $\boldsymbol{y}$. *In such a case, we classify the non-covered query pattern based on the generated rule which has the nearest distance with it. For a non-covered query pattern* $\boldsymbol{y}$, *we first find the fuzzy region* $\boldsymbol{A}^* = (A_1^*, A_2^*, \cdots, A_P^*)$ *that has the greatest matching degree with it. The distance between a non-covered query pattern* $\boldsymbol{y}$ *and one generated rule* R^q *is defined as* $d(\boldsymbol{y}, R^q) = \|\boldsymbol{A}^* - \boldsymbol{A}^q\|_2$, *where* $\boldsymbol{A}^q$ *is the antecedent fuzzy region of rule* R^q.

5.2.3.2 Reasoning using belief functions

In the previous part, the association degrees of the query pattern $\mathbf{y}$ with the consequences of the activated belief rules are calculated. In essence, the association degree is a measure of the reliability of the corresponding consequence regarding the class of the query pattern. Therefore, in the consequence combining process, the reliability of consequence of each activated belief rule should be taken into account. In the framework of belief functions, Shafer's discounting operation, defined by Eq. (1.24), is usually used to discount the unreliable evidence before combination. Regarding association degree α in Eq. (5.19) as the reliability factor, the consequence of one activated belief rule in Eq. (5.5) is discounted

using Shafer's discounting operation as

$$\begin{cases} {}^{\alpha}m(\{\omega_m\}) &= \alpha\beta_m, \quad m = 1, 2, \cdots, M \\ {}^{\alpha}m(\Omega) &= \alpha\beta_\Omega + (1-\alpha). \end{cases} \tag{5.20}$$

For all of the $|\mathcal{S}'| = L$ activated belief rules, with the above formula, we can get the corresponding discounted consequences ${}^{\alpha}m_i$, $i = 1, 2, \cdots, L$.

To make a decision regarding the discounted consequences of activated belief rules, the corresponding mass functions can be combined using Dempster's rule. However, as indicated in [32,123], the direct use of Dempster's rule will result in an exponential increase in computational complexity for the reason of enumerating all subsets or supersets of a given subset A of Ω, and the operation becomes impractical when the frame of discernment has more than 15 to 20 elements. The following part is intended to develop an operational algorithm for evidence combination with linear computational complexity, considering the fact that the focal elements of each associated mass function are all singletons except the ignorance set Ω.

Define $I(i)$ as the index set of the former i mass functions. Let $m_{I(i)}$ be the mass function after combining all of the former i mass functions associated with $I(i)$. Given the above definitions, a recursive evidence combination algorithm can be developed as follows

$$\begin{aligned} m_{I(i+1)}(\{\omega_q\}) &= K_{I(i+1)} \left[m_{I(i)}(\{\omega_q\})^{\alpha}m_{i+1}(\{\omega_q\}) + m_{I(i)}(\Omega)^{\alpha}m_{i+1}(\{\omega_q\}) \right. \\ &\quad \left. + m_{I(i)}(\{\omega_q\})^{\alpha}m_{i+1}(\Omega) \right], \quad q = 1, 2, \cdots, M, \\ m_{I(i+1)}(\Omega) &= K_{I(i+1)} \left[m_{I(i)}(\Omega)^{\alpha}m_{i+1}(\Omega) \right], \\ K_{I(i+1)} &= \left[1 - \sum_{j=1}^{M} \sum_{p=1, p\neq j}^{M} m_{I(i)}(\{\omega_j\})^{\alpha}m_{i+1}(\{\omega_p\}) \right]^{-1} \\ &\quad i = 1, 2, \cdots, L-1, \end{aligned} \tag{5.21}$$

where $K_{I(i+1)}$ is a normalizing factor, so that $\sum_{q=1}^{M} m_{I(i+1)}(\{\omega_q\}) + m_{I(i+1)}(\Omega) = 1$.

Note that $m_{I(1)}(\{\omega_q\}) = {}^{\alpha}m_1(\{\omega_q\})$ for $q = 1, 2, \cdots, M$, and $m_{I(1)}(\Omega) = {}^{\alpha}m_1(\Omega)$. Thus, this recursive evidence combination algorithm can initiate with the first mass function. Accordingly, as the recursive index i reaches $L-1$, the final results $m_{I(L)}(\{\omega_q\})$, $q = 1, 2, \cdots, M$ and $m_{I(L)}(\Omega)$ ($m(\{\omega_q\})$ and $m(\Omega)$ for short, respectively) are obtained by combining all of the L mass functions. This combination result is the basis for the later decision process.

For decision making based on the combined mass function m calculated with Eq. (5.21), the belief function Bel, plausibility function Pl and pignistic probability $BetP$ are common alternatives. As the focal elements of the combined mass function m are all singletons except the ignorance set Ω, the credibility, plausibility and pignistic probability of each

class ω_q are calculated as follows

$$\begin{array}{lcl} Bel(\{\omega_q\}) & = & m(\{\omega_q\}), \\ Pl(\{\omega_q\}) & = & m(\{\omega_q\}) + m(\Omega), \\ BetP(\{\omega_q\}) & = & m(\{\omega_q\}) + \dfrac{m(\Omega)}{M}, \\ & & q = 1, 2, \cdots, M. \end{array} \tag{5.22}$$

It is supposed that based on this evidential body, a decision has to be made in assigning query pattern $\mathbf{y}$ to one of the classes in Ω. Because of the particular structure of the combined mass function (i.e., the focal elements are either singletons or the whole frame Ω), it can be easily discovered that

$$\begin{array}{rcl} \omega & = & \arg\max\limits_{\omega_q \in \Omega} Bel(\{\omega_q\}) \\ & = & \arg\max\limits_{\omega_q \in \Omega} Pl(\{\omega_q\}) \\ & = & \arg\max\limits_{\omega_q \in \Omega} BetP(\{\omega_q\}) \\ & = & \arg\max\limits_{\omega_q \in \Omega} m(\{\omega_q\}). \end{array} \tag{5.23}$$

That is, the strategies maximizing the three criteria Bel, Pl, and $BetP$ in Eq. (5.22) lead to the same decision: the query pattern is assigned to the class with maximum basic belief assignment.

Remark 5.6. *For some classification applications under harsh working conditions (e.g., battlefield target recognition), significant noise may exist in the training set. Though the consequence generation method proposed in Section 5.2.2.1 can reduce the adverse effects from pattern noise, the consequence of one rule may still be unreliable in excessively noisy conditions. The BRM developed within the framework of belief functions combines the consequences of all of the activated rules to obtain the final consequent class. Therefore, compared with the single winner FRM, the BRM can further reduce the risk of misclassification.*

5.3 Experiments

The performance of the proposed BRBCS was empirically assessed by three different experiments with 20 real-world classification problems from the well-known UCI Repository of Machine Learning Databases [72]. In the first experiment, the original data sets were used to evaluate the classification accuracy of the proposed BRBCS. In the second one, the noise was added to the data sets artificially in controlled settings to evaluate the classification robustness of the proposed BRBCS in noisy training set conditions. In the last experiment, we provided an analysis for its time complexity.

5.3.1 Data sets and experimental conditions

Twenty well-known benchmark data sets from the UCI repository were selected to evaluate the performance of the BRBCS. The main characteristics of the 20 data sets are summarized in Table 5.1, where "# Instances" is the number of instances in the data set, "# Features" is the number of features, and "# Classes" is the number of classes. Notice that for the data sets *Cancer*, *Diabetes* and *Pima*, we have removed the instances with missing feature values.

Table 5.1: Description of the benchmark data sets employed in the study

Data set	# Instances	# Features	# Classes
Banknote	1,372	4	2
Breast	106	9	6
Cancer[a]	683	9	2
Diabetes[a]	393	8	2
Ecoli	336	7	8
Glass	214	9	6
Haberman	306	3	2
Iris	150	4	3
Knowledge	403	5	4
Letter	20,000	16	26
Liver	345	6	2
Magic	19,020	10	2
Pageblocks	5,473	10	4
Pima[a]	336	8	2
Satimage	6,435	36	6
Seeds	210	7	3
Transfusion	748	4	2
Vehicle	846	18	4
Vertebral	310	6	3
Yeast	1,484	8	10

[a]For the data sets containing missing values, instances with missing feature values are removed.

To develop the different experiments, we considered the *B-Fold Cross-Validation* (B-CV) model [85]. Each data set was divided into B blocks, with $B-1$ blocks as a training set and the remaining block as a test set. Therefore, each block was used exactly once as a test set. We used the 5-CV here, i.e., five random partitions of the original data set, with four of them (80%) as the training set and the remainder (20%) as the test set. For each data set, we considered the average results of the five partitions.

For the first and the third experiments, the original data sets described above were used directly, whereas for the second, some additional processes were needed. As discussed

in [91, 92, 136], the pattern noise in the data set can be distinguished into two categories: class noise and feature noise. The class noise, also known as labeling error, occurs when a sample is assigned to an incorrect class. It can be attributed to several causes, including subjectivity during the labeling process, data entry errors, or limitations of the equipped measure instrument. In contrast, the feature noise is used to refer to corruptions in the values of one or more features of samples in a data set, which is often encountered in harsh working conditions. With the above consideration, in the second experiment, we managed the robustness evaluation under two types of noise scenarios, class noise and feature noise.

As the initial amount of noise present in the original data sets was unknown, we used manual mechanisms to independently add noise to each data set to control the noise level for comparison. Additionally, to observe how noise affects the accuracy of the classifiers, the noise was only added in the training sets, and the test sets remained unchanged. Based on the type of noise, as in [91], different schemes of noise introduction were designed as follows.

- *Introduction of class noise.* In this scheme, a class noise level of $x\%$ indicates that $x\%$ of the samples in the training set are mislabeled. The class labels of these samples are randomly changed to different ones within the domain of the class.

- *Introduction of feature noise.* In this scheme, a feature noise level of $x\%$ indicates that $x\%$ of the feature values in the training set are erroneous. The corrupted feature is assigned a random value between the minimum and maximum of the domain of that feature, following a uniform distribution.

To evaluate the performance of the difference methods, in the first experiment, the classification accuracy criterion was used. In the second experiment, apart from the classification accuracy under each level of induced noise, we also taken into account the following *relative loss of accuracy* (RLA) to observe the form in which the accuracy of one algorithm was affected when increasing the level of noise with respect to the case without noise.

$$RLA_{x\%} = \frac{Acc_{0\%} - Acc_{x\%}}{Acc_{0\%}}, \tag{5.24}$$

where $RLA_{x\%}$ is the relative loss of accuracy at noise level $x\%$, $Acc_{0\%}$ is the classification accuracy in the test with the original data set, and $Acc_{x\%}$ is the classification accuracy when testing the data set with noise level $x\%$.

To assess whether significant differences exist among different methods, we adopted a nonparametric statistical analysis. For conducting multiple statistical comparisons over multiple data sets, as suggested in [25,41], the Friedman test and the corresponding *post hoc* Bonferroni-Dunn test were employed. For performing multiple comparisons, it is necessary to check whether the results obtained by different methods present any significant difference

(Friedman test), and in the case of finding one, we can find out by using a *post hoc* test to compare the control method with the remaining methods (Bonferroni-Dunn test). We used $\alpha = 0.05$ as the level of significance in all cases. For a detailed description of these tests, one can refer to [25, 41].

5.3.2 Classification accuracy evaluation

In the first experiment, we aim to compare the classification accuracy of our proposed BRBCS with the classical FRBCS proposed by Chi et al. [17], and two improved FRBCSs (denoted as EFRBCS [19] and EBRB [64], respectively) reviewed in Section 2.4.2. The settings of the considered methods are summarized in Table 5.2. As for the considered data sets no prior knowledge about the establishment of the fuzzy regions was available, and the fuzzy grids were used to partition the feature space. We normalized each feature value into a real number in the unit interval $[0, 1]$. Once the number of partitions for each feature was determined, the fuzzy partitions can be easily computed. Here, different numbers of partitions ($C = 3, 5, 7$) were employed to make the comparison.

Table 5.2: Settings of considered methods for classification accuracy evaluation

Method	**Setting**			
	Rule structure	Reasoning method	Membership function	Partition number
FRBCS	Eq. (2.10)	Single winner	Triangular	C = 3,5,7
EFRBCS	Eq. (2.13)	Additive combination	Triangular	C = 3,5,7
EBRB	Eq. (2.15)	Additive combination	Triangular	C = 3,5,7
BRBCS	Eq. (5.1)	Belief reasoning	Triangular	C = 3,5,7

Table 5.3 show the classification accuracy of our proposed BRBCS in comparison with other rule-based methods over the test data. The numbers in brackets represent the rank of each method. It can be seen that, the proposed BRBCS outperforms other methods for most of the data sets. To compare the results statistically, we used nonparametric tests for multiple comparisons to find the best method, considering the average ranks obtained over the test data. First, we used the Friedman test to determine whether significant differences exist among all of the mean values. Table 5.4 shows the Friedman statistic $\mathcal{F}_F$ for each number of partitions, and it relates them to the corresponding critical values by using a level of significance of $\alpha = 0.05$. Given that the Friedman statistics are clearly greater than their associated critical values, there are significant differences among the observed results with a level of significance $\alpha = 0.05$ for all of the three partition numbers. Then, we applied the Bonferroni-Dunn test to compare the best ranking method (i.e., BRBCS) with the remaining methods. Table 5.5 presents these results. We can see that the Bonferroni-Dunn test rejects all of the hypotheses of equality with the rest of the methods with $p < \alpha/(k-1)$. Therefore, by the analysis of the statistical study shown in Tables 5.4 and

Table 5.3: Classification accuracy rate (in %) of our proposed BRBCS in comparison with other rule-based methods for different numbers of partitions

	C = 3				C = 5				C = 7			
	FRBCS	EFRBCS	EBRB	BRBCS	FRBCS	EFRBCS	EBRB	BRBCS	FRBCS	EFRBCS	EBRB	BRBCS
Banknote	94.23(4)	95.33(3)	98.16(1)	96.42(2)	94.53(4)	97.59(1)	96.13(3)	97.15(2)	99.05(3)	99.64(2)	97.08(4)	99.71(1)
Breast	58.57(4)	66.10(2)	65.33(3)	68.33(1)	62.38(4)	67.38(3)	69.24(2)	73.57(1)	59.52(4)	63.33(3)	69.24(2)	70.38(1)
Cancer	90.00(4)	90.44(3)	92.44(2)	95.82(1)	91.82(3)	92.00(2)	91.47(4)	96.74(1)	89.32(4)	91.82(3)	93.59(1)	92.47(2)
Diabetes	67.95(4)	68.56(2)	68.21(3)	69.67(1)	73.08(3)	71.54(4)	73.21(2)	77.82(1)	75.28(2)	75.44(1)	73.67(4)	74.05(3)
Ecoli	76.12(3)	77.79(2)	71.49(4)	78.34(1)	86.57(2)	82.39(3)	81.19(4)	88.06(1)	85.16(2)	84.00(4)	84.49(3)	86.57(1)
Glass	66.05(3)	61.38(4)	66.67(2)	69.04(1)	72.94(1)	68.57(3)	67.00(4)	71.84(2)	67.14(1)	64.57(2)	63.29(4)	64.29(3)
Haberman	67.13(4)	71.80(2)	72.79(1)	68.85(3)	69.18(3)	71.80(2)	62.95(4)	72.46(1)	70.16(3)	71.80(2)	63.77(4)	73.44(1)
Iris	92.67(4)	93.00(3)	95.33(1)	93.67(2)	95.33(2)	94.67(3)	92.00(4)	96.33(1)	96.33(2)	95.67(3)	90.00(4)	96.67(1)
Knowledge	83.25(3)	80.25(4)	86.75(2)	87.25(1)	91.00(3)	82.75(4)	92.75(2)	93.75(1)	83.50(3)	85.00(1)	80.25(4)	84.75(2)
Letter	92.05(3)	94.44(2)	91.92(4)	95.60(1)	90.50(4)	94.12(2)	92.86(3)	95.15(1)	89.32(4)	91.46(3)	93.68(1)	93.00(2)
Liver	56.52(3)	55.65(4)	60.29(1)	59.42(2)	63.07(3)	64.87(2)	58.84(4)	66.52(1)	66.39(3)	66.84(2)	60.00(4)	68.22(1)
Magic	79.55(4)	82.75(2)	82.12(3)	84.96(1)	82.44(2)	81.38(3)	81.06(4)	85.32(1)	76.28(4)	79.45(3)	81.55(2)	82.14(1)
Pageblocks	89.34(4)	95.58(2)	91.63(3)	96.03(1)	90.41(4)	92.67(3)	94.87(2)	95.42(1)	87.89(4)	88.34(3)	90.37(2)	91.67(1)
Pima	64.05(3)	68.10(2)	61.18(4)	69.93(1)	65.36(4)	72.16(2)	71.10(3)	74.71(1)	63.40(4)	66.33(1)	64.10(3)	64.75(2)
Satimage	86.45(4)	87.78(3)	90.65(2)	91.15(1)	82.83(4)	84.36(3)	91.48(1)	89.04(2)	79.38(4)	82.65(3)	89.97(1)	84.56(2)
Seeds	79.52(4)	82.38(3)	85.90(2)	87.00(1)	88.57(3)	90.00(2)	84.76(4)	90.48(1)	86.67(3)	87.30(2)	81.90(4)	88.57(1)
Transfusion	71.81(4)	76.24(2)	75.57(3)	76.51(1)	77.84(2)	76.24(3)	71.68(4)	80.84(1)	78.52(2)	77.72(3)	73.47(4)	83.89(1)
Vehicle	60.36(4)	64.50(3)	69.64(2)	70.91(1)	60.36(4)	66.44(3)	67.45(2)	68.25(1)	57.99(4)	62.72(3)	65.99(2)	66.30(1)
Vertebral	67.42(4)	72.90(2)	69.03(3)	73.87(1)	82.26(3)	83.74(2)	77.29(4)	86.77(1)	81.29(2)	79.71(3)	78.06(4)	84.84(1)
Yeast	48.51(3)	47.30(4)	49.32(2)	56.62(1)	56.32(3)	57.77(2)	52.70(4)	58.53(1)	55.81(1)	53.73(4)	53.95(3)	54.05(2)
Av. Rank	3.65	2.70	2.40	1.25	3.05	2.60	3.20	1.15	2.95	2.55	3.00	1.50

5.5, we conclude that our BRBCS is a solid model for classifier design, as it has shown itself to be the best accuracy method when compared with the other rule-based methods applied in this study.

Table 5.4: Friedman test of the accuracy for the considered methods ($\alpha = 0.05$)

Partition number	Statistic $\mathcal{F}_F$	Critical value	Hypothesis
C = 3	27.005	2.490	Rejected
C = 5	21.000	2.490	Rejected
C = 7	7.798	2.490	Rejected

Table 5.5: Bonferroni-Dunn test of the accuracy for comparing BRBCS with other methods ($\alpha = 0.05$)

Partition number	Method	z value	p value	Critical value $\alpha/(k-1)$[a]	Hypothesis
C = 3	FRBCS	5.88	4.13E-9	0.0167	Rejected
	EFRBCS	3.55	3.83E-4	0.0167	Rejected
	EBRB	2.82	0.0048	0.0167	Rejected
C = 5	FRBCS	4.65	3.26E-6	0.0167	Rejected
	EFRBCS	3.55	3.83E-4	0.0167	Rejected
	EBRB	5.02	5.13E-7	0.0167	Rejected
C = 7	FRBCS	3.55	3.83E-4	0.0167	Rejected
	EFRBCS	2.57	0.0101	0.0167	Rejected
	EBRB	3.67	2.39E-4	0.0167	Rejected

[a] k is the number of considered methods.

To analyze the effect of partition numbers on classification performance, in Table 5.3, the best accuracy for each data set is underlined. It can be seen that the classification accuracy is not always ideally improving according to the increase of partition number, especially for those data sets with relatively more features, which is caused by the limited number of training samples. Additionally, as will be shown in Section 5.3.4, a larger partition number usually means a greater computation burden. Therefore, in practice, for those data sets with fewer features ($M < 10$), we suggest using a partition number $C = 5$; otherwise, a partition number $C = 3$ is suggested to get a better trade-off between accuracy and complexity.

5.3.3 Classification robustness evaluation

In the second experiment, we aim to analyze the classification robustness of our proposed BRBCS when noise is present in the training sets. Apart from the classical FRBCS [17] introduced in Section 2.4.1, the following two robust classifiers were also considered for comparison.

1. *C4.5* [83]: C4.5 is considered to be a robust learner tolerant to noisy data. It iteratively builds a decision tree that correctly classifies the largest number of examples. Additionally, a pruning strategy is used to reduce the chances of the classifier being affected by noisy data from the training set.

2. *BagC4.5* [91]: This is a multiple classifier system that considers C4.5 as the base classifier. In this method, the bagging technique is used to resample the original training set, and then the base classifier is trained with different data sets. As experimentally analyzed in [91], BagC4.5 is a good noise-robust multiple classifier system.

The settings of the considered methods are summarized in Table 5.6. In this experiment, different types of noise (class noise and feature noise) with different noise levels ($NL = 10\%, 20\%, 30\%, 40\%, 50\%$) were tested for comparison.

Table 5.6: Settings of considered methods for classification robustness evaluation

Method	**Setting**
FRBCS	• C = 5 for feature number $M < 10$, otherwise C = 3
C4.5	• Confidence level $c = 0.25$; • Minimal instances per leaf $i = 2$
BagC4.5	• Replicate number $T = 10$; • Majority vote combination
BRBCS	• C = 5 for feature number $M < 10$, otherwise C = 3

Figure 5.2 shows the classification accuracy of each data set at different class noise levels. It may be observed that for most data sets, the proposed BRBCS outperforms the other methods at any class noise level. To verify the robustness of the proposed method more specifically, Table 5.7 gives the RLA of our proposed BRBCS in comparison with other robust methods at different class noise levels. The numbers in brackets represents the rank of each method. For nonparametric statistical analysis, firstly, based on the average ranks of the different methods in Table 5.7, the Friedman test was conducted to evaluate whether significant differences exist among the different methods. Table 5.8 shows the Friedman test result of RLA for the considered methods at different class noise levels. Given that the Friedman statistics are clearly greater than their associated critical values, there are significant differences among the observed results with a level of significance of $\alpha = 0.05$ at all class noise levels. Then, we used the Bonferroni-Dunn test to compare the best ranking method (i.e., BRBCS) with the remaining methods. Table 5.9 presents these results. We can see that the Bonferroni-Dunn test rejects all of the hypotheses of equality with the rest of the methods with $p < \alpha/(k-1)$, except for the BagC4.5 method at noise level $NL = 10\%$. This means that there is no significant difference only between BRBCS and BagC4.5 at noise level $NL = 10\%$ with significance level $\alpha = 0.05$. With the increase of the noise level, the p value associated with each of the remaining methods becomes much

lower. Thus, the RLA differences are more significant at higher noise levels, showing the superior robustness of the proposed BRBCS in disruptive class noise conditions.

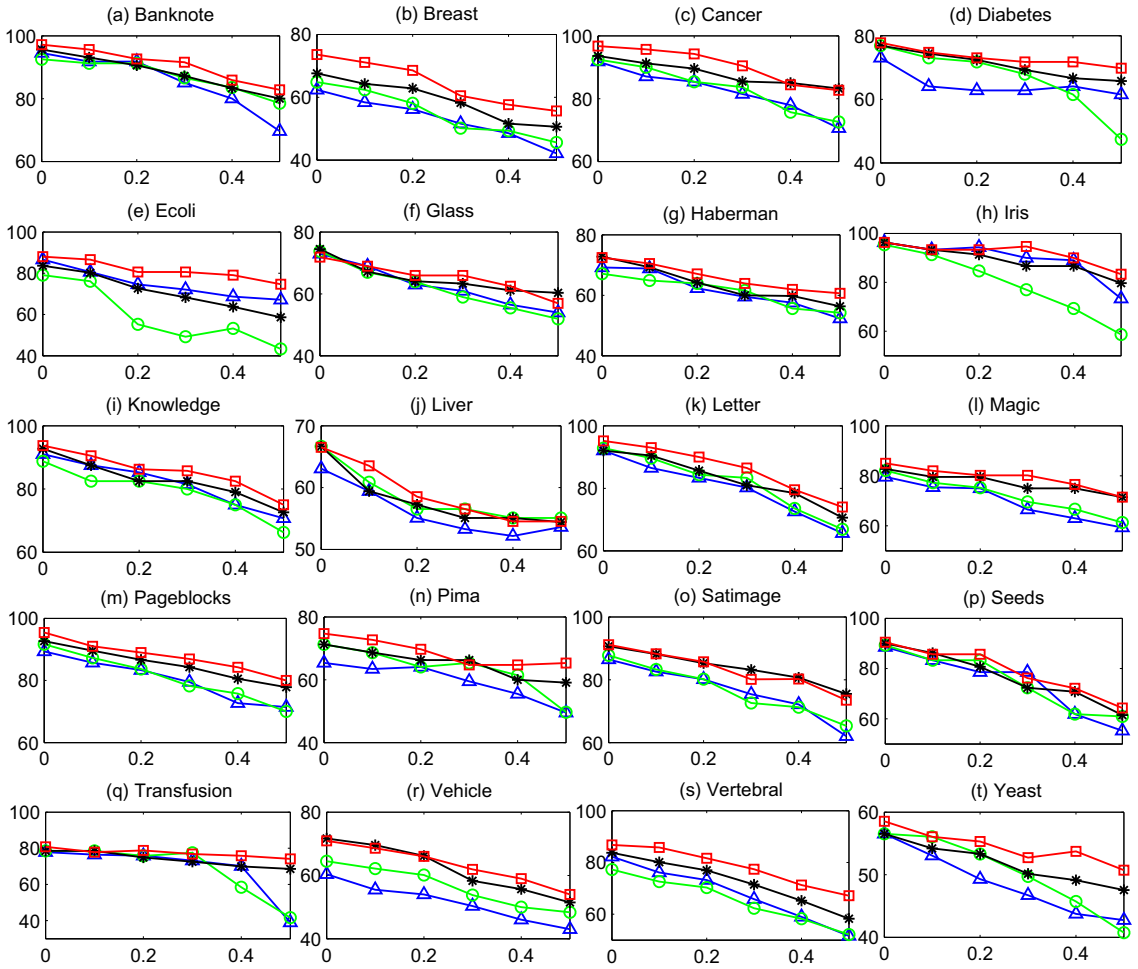


Figure 5.2: Classification accuracy rate (in %) of our proposed BRBCS in comparison with other methods at different class noise levels (The symbol $'\triangle'$ denotes the FRBCS, $'\circ'$ denotes the C4.5, $'*'$ denotes the BagC4.5, and $'\square'$ denotes the BRBCS.)

Figure 5.3 shows the classification accuracy of each data set with different feature noise levels. Similar to the results under class noise conditions, for most data sets, the test accuracy is always higher for BRBCS than for the other robust methods under the feature noise scheme. Table 5.10 shows the RLA of our proposed BRBCS in comparison with other robust methods at different feature noise levels. In a similar manner, we first used the Friedman test to evaluate whether significant differences exist among the different methods. Table 5.11 shows the Friedman test result of RLA for the considered methods at different feature noise levels. It can be seen that the Friedman statistics are clearly greater than their associated critical values at all feature noise levels, which means that there are significant differences among the observed results with a level of significance of $\alpha = 0.05$. Then, we used the Bonferroni-Dunn test to compare the best ranking method

Table 5.7: RLA (in %) of our proposed BRBCS in comparison with other robust methods at different class noise levels

	NL = 10%				NL = 30%				NL = 50%			
	FRBCS	C4.5	BagC4.5	BRBCS	FRBCS	C4.5	BagC4.5	BRBCS	FRBCS	C4.5	BagC4.5	BRBCS
Banknote	2.89(4)	1.40(1)	2.67(3)	1.58(2)	10.04(4)	6.40(2)	8.78(3)	5.71(1)	26.34(4)	15.13(2)	16.32(3)	14.74(1)
Breast	6.41(4)	3.75(2)	4.93(3)	3.34(1)	17.33(2)	22.63(4)	13.87(1)	17.80(3)	32.52(4)	29.61(3)	25.14(2)	24.40(1)
Cancer	5.19(4)	2.64(3)	2.50(2)	1.07(1)	11.27(4)	9.47(3)	8.69(2)	6.48(1)	23.12(4)	21.50(3)	11.10(1)	14.57(2)
Diabetes	12.28(4)	5.00(3)	3.27(1)	3.86(2)	14.04(4)	11.67(3)	9.97(2)	7.74(1)	15.79(3)	38.33(4)	14.50(2)	10.28(1)
Ecoli	6.93(4)	3.77(2)	4.07(3)	1.69(1)	16.71(2)	37.74(4)	18.34(3)	8.47(1)	22.41(2)	45.28(4)	29.83(3)	15.24(1)
Glass	5.48(2)	9.22(3)	9.85(4)	4.08(1)	16.45(3)	20.07(4)	14.98(2)	8.26(1)	26.05(3)	29.56(4)	18.99(2)	17.78(1)
Haberman	1.47(1)	3.34(3)	4.85(4)	2.72(2)	13.98(3)	8.27(1)	17.57(4)	11.77(2)	24.41(4)	19.25(2)	22.61(3)	16.29(1)
Iris	3.11(2)	4.20(4)	3.53(3)	2.89(1)	6.57(2)	19.23(4)	10.03(3)	1.73(1)	23.88(3)	38.46(4)	17.30(2)	13.50(1)
Knowledge	3.85(2)	7.04(4)	5.66(3)	3.47(1)	10.45(3)	9.86(2)	11.05(4)	8.53(1)	22.25(3)	25.35(4)	21.70(2)	20.00(1)
Letter	6.06(4)	3.46(3)	1.74(1)	2.26(2)	12.86(4)	10.26(2)	12.03(3)	9.09(1)	28.69(4)	27.97(3)	23.31(2)	22.23(1)
Liver	5.79(2)	8.70(3)	10.87(4)	4.51(1)	15.53(3)	15.22(2)	17.40(4)	15.03(1)	14.98(1)	17.39(3)	18.37(4)	17.04(2)
Magic	5.22(3)	5.97(4)	3.88(2)	3.49(1)	16.40(4)	15.27(3)	9.41(1)	9.94(2)	25.46(4)	25.38(3)	13.64(1)	15.89(2)
Pageblocks	4.09(2)	4.78(4)	3.33(1)	4.63(3)	10.99(3)	14.60(4)	9.05(2)	8.93(1)	20.04(2)	23.61(4)	20.99(3)	16.16(1)
Pima	3.00(2)	3.67(4)	3.25(3)	2.68(1)	9.00(3)	8.26(2)	6.96(1)	12.10(4)	24.30(3)	30.28(4)	17.06(2)	12.60(1)
Satimage	4.49(3)	5.13(4)	2.76(1)	3.16(2)	12.79(3)	17.30(4)	8.27(1)	12.10(2)	28.28(4)	25.62(3)	16.73(1)	19.40(2)
Seeds	5.91(3)	6.72(4)	4.23(1)	5.26(2)	11.29(1)	19.03(3)	19.58(4)	15.29(2)	37.63(4)	32.77(3)	31.67(2)	28.95(1)
Transfusion	1.71(3)	0.59(2)	0.25(1)	3.71(4)	6.00(2)	8.89(3)	10.85(4)	4.95(1)	49.91(4)	47.01(3)	12.68(2)	8.24(1)
Vehicle	8.10(4)	3.66(3)	2.88(1)	3.26(2)	17.67(3)	16.52(2)	18.62(4)	12.70(1)	28.76(4)	25.07(2)	28.20(3)	23.85(1)
Vertebral	7.45(4)	6.09(3)	4.32(2)	1.11(1)	20.00(4)	19.34(3)	14.68(2)	10.78(1)	37.26(4)	32.54(3)	30.39(2)	22.60(1)
Yeast	6.14(4)	0.83(1)	4.10(2)	4.22(3)	17.34(4)	12.03(3)	11.30(2)	9.91(1)	24.41(3)	27.95(4)	15.89(2)	13.33(1)
Av. rank	3.05	3.00	2.25	1.70	3.05	2.90	2.60	1.45	3.35	3.25	2.20	1.20

Table 5.8: Friedman test of RLA for considered methods at different class noise levels ($\alpha = 0.05$)

Noise level	Statistic $\mathcal{F}_F$	Critical value	Hypothesis
NL = 10%	6.3672	2.490	Rejected
NL = 30%	8.7372	2.490	Rejected
NL = 50%	30.096	2.490	Rejected

Table 5.9: Bonferroni-Dunn test of RLA for comparing BRBCS with other methods at different class noise levels ($\alpha = 0.05$)

Noise level	Method	z value	p value	Critical value $\alpha/(k-1)$[a]	Hypothesis
NL = 10%	FRBCS	3.31	9.44E-4	0.0167	Rejected
	C4.5	3.18	0.0015	0.0167	Rejected
	BagC4.5	1.35	0.1779	0.0167	**Accepted**
NL = 30%	FRBCS	3.92	8.88E-5	0.0167	Rejected
	C4.5	3.55	3.83E-4	0.0167	Rejected
	BagC4.5	2.82	0.0048	0.0167	Rejected
NL = 50%	FRBCS	5.27	1.39E-7	0.0167	Rejected
	C4.5	5.02	5.13E-7	0.0167	Rejected
	BagC4.5	2.45	0.0143	0.0167	Rejected

[a] k is the number of considered methods.

(i.e., BRBCS) with the remaining methods. As shown in Table 5.12, the Bonferroni-Dunn test rejects all of the hypotheses of equality with the rest of the methods with $p < \alpha/(k-1)$, except for the BagC4.5 method at noise levels $NL = 10\%$ and $NL = 30\%$. In other words, there is no significant difference between BRBCS and BagC4.5 at noise levels $NL = 10\%$ and $NL = 30\%$ with significance level $\alpha = 0.05$. This is mainly because the feature noise is not very disruptive at relatively lower noise levels. With the increase of the noise level, the p value associated with each of the remaining methods becomes much lower. Thus, the RLA differences are more significant at higher noise levels, which shows the superior robustness of the proposed BRBCS in disruptive feature noise conditions.

5.3.4 Time complexity analysis

In this section, a time complexity analysis of the proposed BRBCS was provided to show to what extent the runtime depends on factors such as the number of instances, the number of features and the number of partitions. Twenty real-world problems (with the numbers of training instances ranging from 85 to 16,000 and the numbers of features ranging from 3 to 36) shown in Table 5.1 were considered for evaluation. Three different numbers of partitions, $C = 3, 5, 7$, were tested, and the 5-CV model was used to calculate the average runtime. The numerical experiments were executed by MATLAB 7.12 on an HP EliteBook

Table 5.10: RLA (in %) of our proposed BRBCS in comparison with other robust methods at different feature noise levels

	NL = 10%				NL = 30%				NL = 50%			
	FRBCS	C4.5	BagC4.5	BRBCS	FRBCS	C4.5	BagC4.5	BRBCS	FRBCS	C4.5	BagC4.5	BRBCS
Banknote	3.44(3)	2.71(2)	4.06(4)	1.05(1)	8.79(4)	8.53(3)	7.02(2)	4.62(1)	12.48(3)	15.60(4)	11.40(2)	8.27(1)
Breast	5.34(4)	4.48(3)	3.33(2)	2.26(1)	12.34(1)	16.79(3)	20.00(4)	13.91(2)	20.46(2)	21.42(3)	26.67(4)	17.99(1)
Cancer	4.91(3)	5.52(4)	1.17(1)	4.38(2)	8.81(3)	9.63(4)	7.95(1)	8.38(2)	16.22(4)	12.78(2)	15.94(3)	12.44(1)
Diabetes	0.36(1)	6.67(4)	3.92(3)	1.15(2)	7.20(1)	12.23(4)	11.03(3)	9.39(2)	14.04(4)	12.23(2)	13.27(3)	9.39(1)
Ecoli	8.62(3)	9.44(4)	4.69(2)	2.82(1)	13.78(4)	13.21(3)	10.72(2)	5.09(1)	24.14(4)	17.63(3)	17.35(2)	10.18(1)
Glass	5.48(4)	1.08(1)	3.07(3)	1.30(2)	8.23(3)	10.58(4)	7.70(2)	4.08(1)	15.08(3)	16.00(4)	14.03(2)	7.53(1)
Haberman	0.38(1)	6.22(4)	0.79(2)	1.59(3)	11.19(4)	10.15(3)	7.56(2)	7.35(1)	20.62(4)	18.20(3)	11.28(1)	12.90(2)
Iris	0.70(1)	1.77(3)	3.45(4)	1.39(2)	2.45(2)	4.90(3)	5.52(4)	2.08(1)	3.85(2)	9.09(4)	7.59(3)	2.78(1)
Knowledge	3.85(3)	4.69(4)	2.14(1)	2.66(2)	9.26(2)	9.40(3)	9.89(4)	7.99(1)	14.84(1)	16.73(3)	17.15(4)	15.13(2)
Letter	4.00(4)	0.73(1)	2.64(2)	3.31(3)	9.34(3)	8.58(2)	9.44(4)	7.28(1)	16.21(4)	14.36(3)	13.35(2)	11.56(1)
Liver	0.33(1)	2.17(4)	1.55(3)	0.67(2)	5.79(1)	15.22(3)	15.51(4)	7.72(2)	12.68(1)	26.09(4)	20.22(3)	15.68(2)
Magic	3.55(4)	1.77(3)	1.30(2)	1.01(1)	6.82(3)	9.63(4)	3.58(1)	5.27(2)	12.88(2)	13.63(4)	13.45(3)	12.72(1)
Pageblocks	3.86(3)	2.99(2)	4.55(4)	2.73(1)	10.22(3)	12.43(4)	9.48(2)	7.07(1)	14.07(2)	18.98(4)	15.24(3)	11.15(1)
Pima	1.00(1)	3.67(3)	5.22(4)	1.14(2)	4.18(1)	9.28(2)	12.28(4)	9.89(3)	10.70(1)	12.09(2)	19.50(4)	16.17(3)
Satimage	4.85(3)	5.20(4)	0.90(1)	1.02(2)	7.63(3)	11.01(4)	4.74(1)	6.14(2)	14.93(3)	16.03(4)	9.91(1)	10.41(2)
Seeds	4.30(4)	4.08(3)	3.04(2)	2.63(1)	8.60(3)	12.70(4)	5.66(1)	5.98(2)	13.28(3)	16.79(4)	11.92(2)	10.84(1)
Transfusion	1.84(2)	0.85(1)	5.71(4)	5.35(3)	7.05(3)	4.33(1)	7.80(4)	6.19(2)	10.70(3)	9.40(2)	12.02(4)	6.19(1)
Vehicle	4.82(3)	5.29(4)	1.30(1)	2.06(2)	10.40(4)	10.12(3)	7.99(2)	6.47(1)	20.00(4)	19.58(3)	17.34(2)	15.34(1)
Vertebral	2.58(4)	2.13(3)	1.18(1)	1.68(2)	9.02(4)	7.35(3)	6.03(1)	6.95(2)	13.88(4)	10.61(1)	12.40(3)	12.08(2)
Yeast	6.14(4)	5.71(3)	1.94(2)	0.22(1)	17.34(4)	12.03(3)	8.14(2)	5.52(1)	24.41(4)	20.40(3)	14.17(2)	8.20(1)
Av. rank	2.80	3.00	2.40	1.80	2.80	3.15	2.50	1.55	2.90	3.10	2.65	1.35

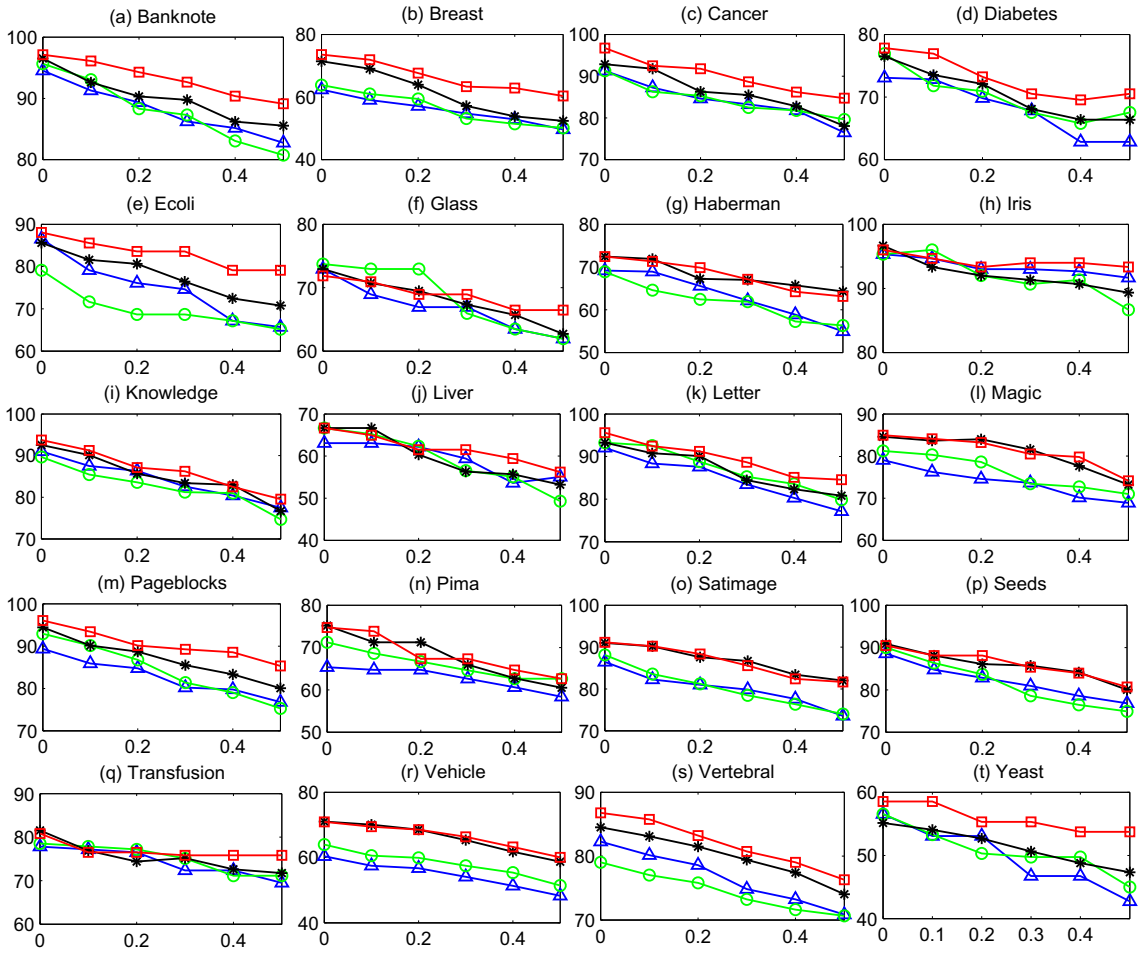


Figure 5.3: Classification accuracy rate (in %) of our proposed BRBCS in comparison with other methods at different feature noise levels (The symbol $'\triangle'$ denotes the FRBCS, $'\circ'$ denotes the C4.5, $'*'$ denotes the BagC4.5, and $'\square'$ denotes the BRBCS.)

Table 5.11: Friedman test of RLA for considered methods at different feature noise levels ($\alpha = 0.05$)

Noise level	Statistic $\mathcal{F}_F$	Critical value	Hypothesis
NL = 10%	3.8365	2.490	Rejected
NL = 30%	7.4993	2.490	Rejected
NL = 50%	11.303	2.490	Rejected

8570p with an Intel(R) Core(TM) i7-3540 M CPU @3.00 GHz and 8 GB memory. Table 5.13 shows the average runtime of the proposed BRBCS in the training and testing phases for different data sets and different partition numbers, where "# Training" is the number of training instances in the data set, "# Features" is the number of features, and "# Rules" is the number of generated rules. "T. Tra." and "T. Tes." are the average runtimes in the training phase and testing phase (classifying one pattern), respectively.

By analyzing the results presented in Table 5.13, we can see that the runtimes for both

Table 5.12: Bonferroni-Dunn test of RLA for comparing BRBCS with other methods at different feature noise levels ($\alpha = 0.05$)

Noise level	Method	z value	p value	Critical value $\alpha/(k-1)$[a]	Hypothesis
NL = 10%	FRBCS	2.45	0.0143	0.0167	Rejected
	C4.5	2.94	0.0033	0.0167	Rejected
	BagC4.5	1.47	0.1416	0.0167	**Accepted**
NL = 30%	FRBCS	3.06	0.0022	0.0167	Rejected
	C4.5	3.92	8.89E-5	0.0167	Rejected
	BagC4.5	2.33	0.0200	0.0167	**Accepted**
NL = 50%	FRBCS	3.80	1.47E-4	0.0167	Rejected
	C4.5	4.29	1.81E-5	0.0167	Rejected
	BagC4.5	3.18	0.0015	0.0167	Rejected

[a] k is the number of considered methods.

the training and testing phases mainly depend on the number of generated rules. More rules usually means more time to train the BRB and also more time to classify a pattern based on the generated BRB. Thus, we can instead analyze how the factors affect the number of rules. First, for each data set, the number of rules always increases with increases in the partition number. However, the number of rules cannot increase indefinitely, as it is constrained by the number of training instances. Second, by comparing different data sets, we can see that a larger number of features usually results in a larger number of rules (e.g., the data sets *Letter* and *Satimage*). However, this tendency is also constrained by the number of training instances. For example, although the data set *Vehicle* has a larger number of features than *Magic*, it has a relatively smaller number of rules under any partition condition, mainly because its number of training instances is quite small. In brief, with the growth in the numbers of partitions and features, the runtime of the proposed BRBCS will increase, but this increase is constrained by the number of available training instances.

5.4 Conclusion

In this chapter, we have extended the traditional FRBCS in the framework of belief functions and developed a belief rule-based classification system (BRBCS) to address uncertain information in complex classification problems. The two components of the proposed BRBCS, i.e., the belief rule base and the belief reasoning method, have been designed specifically by taking into account the possible pattern noise in many real-world data sets. The experiments have shown that the proposed BRBCS achieves better classification accuracy compared with other rule-based methods. Moreover, this method

Table 5.13: Average runtime (in s) of our proposed BRBCS for different data sets and different partition numbers

			C = 3			C = 5			C = 7		
Data sets	# Training	# Features	# Rules	T. Tra.	T. Tes.	# Rules	T. Tra.	T. Tes.	# Rules	T. Tra.	T. Tes.
Banknote	1,098	4	30.6	0.166	1.3E-3	74.4	0.225	2.7E-3	151.2	0.293	5.2E-3
Breast	85	9	26	0.025	2.2E-3	49	0.031	3.3E-3	61.6	0.047	4.3E-3
Cancer	547	9	209	0.181	1.4E-2	267.2	0.209	1.8E-2	323.2	0.284	2.1E-2
Diabetes	315	8	80.2	0.091	5.4E-3	248.8	0.125	1.6E-2	297.8	0.162	1.8E-2
Ecoli	269	7	44.8	0.050	2.1E-3	105.8	0.072	4.4E-3	178.4	0.094	7.1E-3
Glass	172	9	39.8	0.053	2.9E-3	80.4	0.072	5.7E-3	115.8	0.099	8.2E-3
Haberman	245	3	17.2	0.029	6.0E-4	49	0.041	1.4E-3	82.2	0.047	2.4E-3
Iris	120	4	14.4	0.025	6.1E-4	42.6	0.029	1.6E-3	63	0.031	2.3E-3
Knowledge	323	5	61.4	0.033	2.4E-3	120.2	0.041	5.0E-3	154.2	0.059	6.3E-3
Letter	16,000	16	1354.6	7.025	8.4E-2	3593	11.963	2.0E-1	6939	17.708	3.7E-1
Liver	276	6	42.8	0.053	2.4E-3	112.6	0.078	5.7E-3	171.6	0.106	8.2E-3
Magic	15,216	10	346.2	4.034	2.8E-2	1854.8	5.123	5.7E-2	4573.2	6.933	8.2E-2
Pageblocks	4,379	10	55.4	0.642	4.7E-3	162	0.950	1.1E-2	286.6	1.542	2.2E-2
Pima	615	8	104.6	0.149	6.9E-3	190.8	0.231	2.4E-2	340.8	0.303	3.2E-2
Satimage	5,148	36	1443.2	1.857	1.6E-1	2046.6	5.997	2.5E-1	2531.6	6.730	3.1E-1
Seeds	168	7	53.2	0.041	3.2E-3	96.8	0.062	5.9E-3	121.8	0.078	7.7E-3
Transfusion	599	4	12.6	0.081	5.1E-4	28.6	0.112	1.1E-3	56	0.147	2.0E-3
Vehicle	677	18	288.2	0.374	5.2E-2	334	0.577	8.0E-2	370.8	0.633	8.5E-2
Vertebral	248	6	34	0.049	1.9E-3	93.2	0.066	4.7E-3	142.2	0.094	6.9E-3
Yeast	1,188	8	96	0.312	6.6E-3	208	0.515	1.3E-2	462	0.608	2.8E-2

can effectively address the class or feature noise in the training data set. This allows us to conclude that the introduction of belief functions improves the behavior of the rule-based classification system.

Hybrid belief rule-based classification system

In some real-world pattern classification applications, both training data collected by sensors and expert knowledge may be available. In this chapter, a hybrid belief rule-based classification system (HBRBCS) is developed to make use of these two types of information jointly. The belief rule structure, which is capable of capturing fuzzy, imprecise, and incomplete causal relationships, is used as the common representation model. With the belief rule structure, a data-driven belief rule base (DBRB) and a knowledge-driven belief rule base (KBRB) are learnt from uncertain training data and expert knowledge, respectively. A fusion algorithm is proposed to combine the DBRB and KBRB to obtain an optimal hybrid belief rule base (HBRB), based on which a query pattern is classified by taking into the uncertain information from both training data and expert knowledge.

In Section 6.1, we first describe the background and motivations. The details of the proposed hybrid belief rule-based classification system are presented in Section 6.2. An airborne target classification problem in the air surveillance system is studied in Section 6.3 to demonstrate the performance of the proposed method for combining both uncertain training data and expert knowledge to make classification. Finally, Section 6.4 concludes this chapter.

6.1 Introduction

In the previous chapter, we have developed a belief rule-based classification system to learn from uncertain training data in complex classification problems. However, in some real-world pattern classification applications, apart from the training data, some expert knowledge from humans may also be available. As discussed in Section 2.4.3, these two types of information are usually independent and complementary, and both are useful for classification. Therefore, there is a need for an effective modeling method that can make good use of both training data and expert knowledge, and integrate the best aspects of these two types of information for classification.

In order to combine training data and expert knowledge for classification, a common representation model that can make use of both types of information is needed. The IF-THEN rule is a good representation model because, on the one hand, the IF-THEN rules can be learnt from training data and, on the other hand, expert knowledge is also easily coded into IF-THEN rules. However, different types of uncertainty may coexist in real-world applications, e.g., both training data and expert knowledge may be imprecise or incomplete. In this chapter we use the belief rule as a common model to represent uncertain training data and expert knowledge.

Based on the belief rule structure, a hybrid belief rule-based classification system (HBRBCS) is developed to make good use of these two types of information. The proposed HBRBCS is composed of two main components: a hybrid belief rule base that establishes the association between the feature space and the class space, and a belief reasoning method that provides a mechanism to classify a query pattern based on the constructed rule base. With the belief rule structure, a data-driven belief rule base and a knowledge-driven belief rule base are learnt from uncertain training data and expert knowledge, respectively. A fusion algorithm is proposed to combine the data-driven and knowledge-driven belief rule bases to obtain an optimal hybrid belief rule base. Then, the belief reasoning method is applied to classify a query pattern in a robust way. Finally, we apply the proposed HBRBCS to solve an airborne target classification problem based on uncertain sensor measurements and the expert knowledge.

6.2 Hybrid belief rule-based classification system

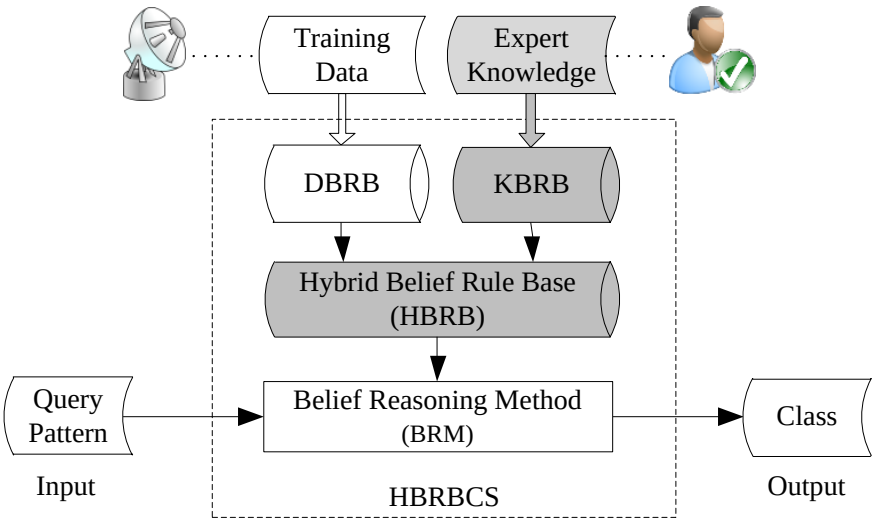


Figure 6.1: Hybrid belief rule-based classification system

As shown Figure 6.1, the proposed HBRBCS is composed of two components: a hybrid

belief rule base and a belief reasoning method. Compared with the BRBCS developed in Chapter 5, two modules are added, i.e., a knowledge-driven belief rule base (KBRB), and a hybrid belief rule base (HBRB). As the construction of data-driven belief rule base (DBRB) and the belief reasoning method have already been developed in Chapter 5, in this section, we focus on how to learn a KBRB from expert knowledge with the belief rule structure and how to obtain an optimal HBRB by combining the DBRB and KBRB.

6.2.1 Knowledge-driven belief rule base

In knowledge-based system development, *knowledge representation* is the task of encoding expert knowledge into a knowledge base. Based on different types of applications, many knowledge representation schemes have been proposed, such as logical representations, production rules, semantic networks and structured frames [43]. For production rules, as knowledge is represented in the form of condition/action pairs, they provide a natural way to characterize the association between feature space and class space for classification problems. Thus, in this section, we select the production rules (specially, the fuzzy IF-THEN rules) to represent expert knowledge.

Based on the fuzzy IF-THEN rules representation, the expert knowledge is acquired from experts using the structured interview technique [71]. That is, experts are asked to assign fuzzy regions to each class and to give corresponding certainty grades. Accordingly, for an M-class (denoted as $\Omega \triangleq \{\omega_1, \omega_2, \cdots, \omega_M\}$) pattern classification problem with P features, each piece of expert knowledge $\mathbf{e}_j$ can be represented as

$$\begin{aligned} &\text{Expert Knowledge } \mathbf{e}_j: \\ &\text{If } x_1 \text{ is } \mathbf{A}_1^j \text{ and } x_2 \text{ is } \mathbf{A}_2^j \text{ and } \cdots \text{ and } x_P \text{ is } \mathbf{A}_P^j, \text{ then consequence is } \omega_j, \\ &\text{with certainty grade } \theta_j, j = 1, 2, \cdots, M, \end{aligned} \tag{6.1}$$

where $\mathbf{A}_p^j$ is subset of fuzzy partitions $\{A_{p,1}, A_{p,2}, \cdots, A_{p,n_p}\}$ associated with the p-th feature, $p = 1, 2, \cdots, P$.

With the above M pieces of expert knowledge $\mathbf{e}_j$, $j = 1, 2, \cdots, M$, the problem now is how to generate a belief rule base with the belief rule structure as Eq. (5.1) from the expert knowledge. The knowledge-driven belief rule base is constructed in the following two stages: first, we expand the expert knowledge $\mathbf{e}_j$, $j = 1, 2, \cdots, M$, into belief rules by enumerating all of the possible antecedent conditions; then, we combine expanded rules having the same antecedent part.

6.2.1.1 Expansion of expert knowledge

For the belief rule as Eq. (5.1), each feature is associated with a single fuzzy partition, whereas for the expert knowledge in Eq. (6.1), each feature is associated with a set of

fuzzy partitions. Thus, one piece of expert knowledge $\mathbf{e}_j$ can be expanded into some belief rules with the same consequent part and rule weight by enumerating all of the possible antecedent conditions as follows.

$$\begin{aligned}
&\text{Belief Rule } R_j^1: && \text{If } x_1 \text{ is } A_1^1 \text{ and } x_2 \text{ is } A_2^1 \text{ and} \cdots \text{ and } x_P \text{ is } A_P^1, \text{ then consequence is} \\
& && \mathbf{C}^1 = \{(\omega_j, 1)\}, \text{ with rule weight } \theta_j, \\
& && \vdots \\
&\text{Belief Rule } R_j^q: && \text{If } x_1 \text{ is } A_1^q \text{ and } x_2 \text{ is } A_2^q \text{ and} \cdots \text{ and } x_P \text{ is } A_P^q, \text{ then consequence is} \\
& && \mathbf{C}^q = \{(\omega_j, 1)\}, \text{ with rule weight } \theta_j, \\
& && \vdots \\
&\text{Belief Rule } R_j^{Q_j}: && \text{If } x_1 \text{ is } A_1^{Q_j} \text{ and } x_2 \text{ is } A_2^{Q_j} \text{ and} \cdots \text{ and } x_P \text{ is } A_P^{Q_j}, \text{ then consequence is} \\
& && \mathbf{C}^{Q_j} = \{(\omega_j, 1)\}, \text{ with rule weight } \theta_j,
\end{aligned} \tag{6.2}$$

where the antecedent parts $(A_1^q, A_2^q, \cdots, A_P^q)$, $q = 1, 2, \cdots, Q_j$, are all the possible combinations of different partitions for $\mathbf{A}_1^j, \mathbf{A}_2^j, \cdots, \mathbf{A}_P^j$, and Q_j is the number of belief rules generated from expert knowledge $\mathbf{e}_j$, with $Q_j = \prod_{p=1}^{P} |\mathbf{A}_p^j|$.

In the same way, the M pieces of expert knowledge $\mathbf{e}_j$, $j = 1, 2, \cdots, M$, can be expanded to generate $\sum_{j=1}^{M} Q_j$ belief rules. However, as different classes may overlap in feature space, the belief rules generated from different pieces of expert knowledge may have the same antecedent part but different consequent parts. These rules are in conflict with each other. In the following, we provide a method to combine these conflicting rules considering their rule weights.

6.2.1.2 Combination of conflicting rules

Suppose $R_{j_1}^{q_1}, \cdots, R_{j_{M'}}^{q_{M'}}$ $(2 \leq M' \leq M)$ are M' generated rules with the same antecedent part but different consequent parts $\{(\omega_{j_1}, 1)\}, \cdots, \{(\omega_{j_{M'}}, 1)\}$. In order to generate a compact KBRB, these M' rules should be fused into a new rule. The antecedent part of this new rule keeps the same, and its consequent part is obtained by combining those of the M' conflicting rules.

Each conflicting rule $R_{j_m}^{q_m}$ provides a piece of evidence that supports the class ω_{j_m} as the consequent part of the new fused rule. Considering that this rule has a certainty grade θ_{j_m}, this piece of evidence can be represented by a mass function m^{q_m} verifying:

$$\begin{cases}
m^{q_m}(\{\omega_{j_m}\}) &= \theta_{j_m} \\
m^{q_m}(\Omega) &= 1 - \theta_{j_m} \\
m^{q_m}(A) &= 0, \quad \forall A \in 2^{\Omega} \setminus \{\Omega, \{\omega_{j_m}\}\},
\end{cases} \tag{6.3}$$

where $\Omega = \{\omega_1, \cdots, \omega_M\}$ is the frame of discernment.

In a similar way, M' pieces of evidence $m^{q_1}, \cdots, m^{q_{M'}}$ can be constructed from the M'

conflicting rules. These pieces of evidence are combined using Dempster's rule as follows:

$$\begin{cases} m(\{\omega_{j_m}\}) &= \dfrac{\theta_{j_m}}{1-K}\displaystyle\prod_{r\neq m}(1-\theta_{j_r}), \quad m=1,\cdots,M' \\ m(\Omega) &= \dfrac{1}{1-K}\displaystyle\prod_{r=1}^{M'}(1-\theta_{j_r}), \end{cases} \tag{6.4}$$

where K is the total conflicting belief mass,

$$K = 1-\prod_{r=1}^{M'}(1-\theta_{j_r}) - \sum_{m=1}^{M'}\theta_{j_m}\prod_{r\neq m}(1-\theta_{j_r}). \tag{6.5}$$

In addition, as the weights of the M' conflicting rules have already been considered in the combination process, the weight of the new fused rule is set to 1. Consequently, the M' conflicting rules can be replaced by a new fused rule with the same antecedent part and full weight but a different consequent part as $\{(\omega_{j_1}, m(\{\omega_{j_1}\})), \cdots, (\omega_{j_{M'}}, m(\{\omega_{j_{M'}}\}))\}$. In a similar way, all other sets of conflicting rules can be replaced by a series of new fused rules and then a compact KBRB is obtained to encode the expert knowledge about the classification problem.

6.2.2 Hybrid belief rule base

In the previous section, a KBRB, dependent from the DBRB, is constructed based on expert knowledge. In this section, we aim to fuse these two different belief rule bases into a new hybrid belief rule base for classification. In real-world classification problems, both training data and expert knowledge may be uncertain. The uncertainty of training data comes from measurement noise, data entry errors, or small size of samples. The uncertainty of expert knowledge is mainly due to limited or uncorrect assessment for the considered problem. Consequently, both the DBRB and KBRB only provide partially reliable information for the classification problem. Thus, in order to get a more powerful hybrid belief rule base, we should take into account the weights of these two rule bases, which reflect their different roles in the fusion process.

6.2.2.1 Fusion of DBRB and KBRB

Assuming the DBRB is β ($\beta > 0$) times as important as the KBRB, the weights of these two BRBs are set to $\beta/(1+\beta)$ and $1/(1+\beta)$, respectively. For notational convenience, we write $\lambda = \beta/(1+\beta)$. Accordingly, the weight of the DBRB is λ and the weight of the KBRB is $1-\lambda$ with $0 < \lambda < 1$. The adjustment factor λ plays an important role in adjusting the hybrid decision boundaries. With a large value of λ, the hybrid boundaries tend toward the DBRB boundaries. In contrast, with a small value of λ, the hybrid boundaries tend toward the KBRB boundaries.

With the above defined weights, we now fuse the Q_D data-driven belief rules (R_D^i, $i = 1, 2, \cdots, Q_D$) in the DBRB with the Q_K knowledge-driven belief rules (R_K^j, $j = 1, 2, \cdots, Q_K$) in the KBRB. As illustrated in Figure 6.2, due to the partial information provided by both training data and expert knowledge, the generated rules in both the DBRB and KBRB may only cover partial fuzzy regions. Furthermore, because of the independence between training data and expert knowledge, the fuzzy regions covered by the DBRB and the KBRB may not fully overlap. Thus, the rules in the integrated HBRB can be divided into the following three categories: rules with fuzzy regions only covered by the DBRB, rules with fuzzy regions only covered by the KBRB, and rules with fuzzy regions covered by both the DBRB and KBRB.

Let $\mathcal{S} = \{(i, j) \mid R_D^i$ and R_K^j have the same antecedent part, $i = 1, 2, \cdots, Q_D, j = 1, 2, \cdots, Q_K\}$, $\mathcal{S}_D = \{i \mid (i, j) \in \mathcal{S}, j = 1, 2, \cdots, Q_K\}$ and $\mathcal{S}_K = \{j \mid (i, j) \in \mathcal{S}, i = 1, 2, \cdots, Q_D\}$. The rules in the integrated HBRB are generated as follows.

- The rules with fuzzy regions only covered by the DBRB are generated by assigning the corresponding rules in the DBRB with new weights $\lambda\theta^i$, $i \in \{1, 2, \cdots, Q_D\} \setminus \mathcal{S}_D$;
- The rules with fuzzy regions only covered by the KBRB are generated by assigning the corresponding rules in the KBRB with new weights $(1-\lambda)\theta^j$, $j \in \{1, 2, \cdots, Q_K\} \setminus \mathcal{S}_K$;
- The rules with fuzzy regions covered by both the DBRB and KBRB are generated by assigning the corresponding rules in both the DBRB and KBRB with new weights $\lambda\theta^i + (1 - \lambda)\theta^j$ and new consequences m^{ij} calculated by

$$m^{ij} = {}^{\lambda}m^i \oplus {}^{(1-\lambda)}m^j, \quad (i, j) \in \mathcal{S}, \tag{6.6}$$

  where ${}^{\lambda}m^i$ denotes the discounted mass function for the consequence of the corresponding rule in DBRB with reliability factor λ, ${}^{(1-\lambda)}m^j$ denotes the discounted mass function for the consequence of the corresponding rule in KBRB with reliability factor $(1 - \lambda)$, and $\oplus$ is Dempster's rule of combination.

Proposition 6.1. *The generated HBRB reduces to DBRB and KBRB when the adjustment factor λ takes 1 and 0, respectively.*

Proof. Suppose the adjustment factor $\lambda = 1$. First, for those rules with fuzzy regions only covered by the DBRB, the new assigned weights $\lambda\theta^i = \theta^i$, $i \in \{1, 2, \cdots, Q_D\} \setminus \mathcal{S}_D$. Thus, this category of rules is kept unchanged.

Second, for those rules with fuzzy regions only covered by the KBRB, the new assigned weights $(1 - \lambda)\theta^j = 0$, $j \in \{1, 2, \cdots, Q_K\} \setminus \mathcal{S}_K$. Thus, this category of rules are excluded from the generated HBRB.

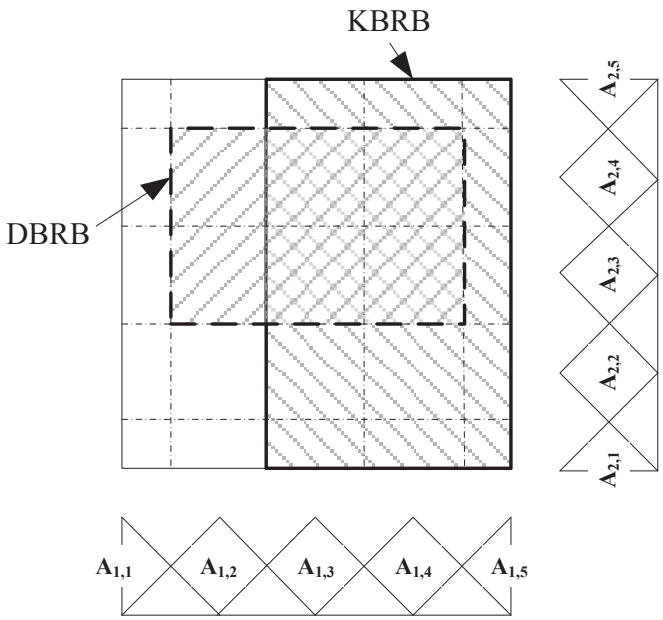


Figure 6.2: An example of the fuzzy regions covered by the DBRB and KBRB for a two-dimensional feature space

Third, for those rules with fuzzy regions covered by both the DBRB and KBRB, the new assigned weights $\lambda\theta^i + (1-\lambda)\theta^j = \theta^i$ and new consequences

$$m^{ij} = {}^{\lambda}m^i \oplus {}^{(1-\lambda)}m^j = {}^{1}m^i \oplus {}^{0}m^j, \quad (i,j) \in \mathcal{S}. \tag{6.7}$$

From the definition of Shafer's discounting operation in Eq. (1.24), it is easy to see that ${}^{1}m^i = m^i$, and ${}^{0}m^j$ becomes to a vacuous mass function. Further, from the definition of Dempster's rule of combination in Eq. (1.10), the combination of any mass function with a vacuous mass function is equal to itself. Therefore, the new consequences $m^{ij} = m^i$, $(i,j) \in \mathcal{S}$. Thus, those rules in overlapping fuzzy regions inherit directly from the corresponding rules in the DBRB.

Consequently, it can be concluded that the generated HBRB reduces to DBRB when the adjustment factor $\lambda = 1$. In a similar way, we can prove that the generated HBRB reduces to KBRB when the adjustment factor $\lambda = 0$. □

6.2.2.2 Optimization of the adjustment factor

In the above HBRB generation process, the rules from the DBRB and KBRB are combined to get an integrated HBRB that can make use of the information from both training data and expert knowledge. In this combination process, the adjustment factor λ is used to adjust the weights of these two types of information. The adjustment factor λ can be specified by the user by evaluating the relative reliability of these two types of information. However, due to the ignorance about the quality of training data or expert knowledge, it

may be difficult for the user to specify a proper value for λ. In the following, we propose to search an optimal value for λ by optimizing the average leave-one-out test error.

Let us consider a training sample $\mathbf{x}_i$ belonging to class ω_m. Take $\mathbf{x}_i$ as a test sample, and $\mathcal{T}_i = \mathcal{T} \setminus \{(\mathbf{x}_i, \omega_m)\}$ as the new training set. A HBRB can be generated based on the new training set $\mathcal{T}_i$ and the expert knowledge. Using the belief reasoning method developed in Chapter 5, one can get the leave-one-out test output $\mathbf{P}_i = (BetP_i(\{\omega_1\}), \cdots, BetP_i(\{\omega_M\}))$. Ideally, the classification output vector $\mathbf{P}_i$ should be as close as possible to the real class vector $\mathbf{t}_i = (t_{i1}, \cdots, t_{iM})$ (each binary indicator variable t_{ij} is defined by $t_{ij} = 1$, if $j = m$ and $t_{ij} = 0$, otherwise), with closeness being defined according to the following squared error:

$$E^{\lambda}(\mathbf{x}_i) = (\mathbf{P}_i - \mathbf{t}_i)(\mathbf{P}_i - \mathbf{t}_i)^{\mathrm{T}} = \sum_{j=1}^{M} (BetP_i(\{\omega_j\}) - t_{ij})^2. \tag{6.8}$$

The mean squared error over the whole training set $\mathcal{T}$ of size N is finally equal to

$$E^{\lambda} = \frac{1}{N} \sum_{i=1}^{N} E^{\lambda}(\mathbf{x}_i). \tag{6.9}$$

Therefore, the optimal value for λ is chosen with minimum leave-one-out test error, i.e.,

$$\hat{\lambda} = \arg \min_{0 \leq \lambda \leq 1} E^{\lambda}. \tag{6.10}$$

As the minimization of E^{λ} is performed with respect to a single parameter in a bounded domain, a simple search procedure can be used.

6.3 Numerical study

In order to present the implementation of the proposed HBRBCS and demonstrate its capacity of combining both uncertain training data and expert knowledge for classification, we provide a numerical study for an airborne target classification in the air surveillance system.

6.3.1 Problem description

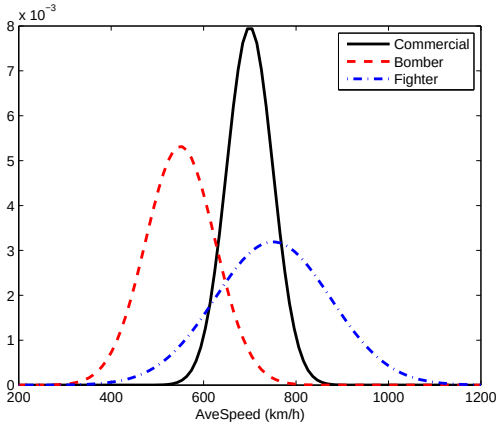
For air surveillance systems [87], one of the most important tasks is to correctly recognize noncooperative flying objects within their surveillance volume. In general, target classification is based on a set of features or attributes that distinguish targets according to their shapes or kinematic behaviors. To fully exploit the feature space, a surveillance system often consists of multiple sensors. For example, a radar can provide kinematic features (e.g., speed, acceleration) and an infrared sensor can supply shape features such as the length. In this study, we consider the classification of targets in predefined categories

$\{Commercial\ plane, Bomber, Fighter\}$, based on their average speed ($AveSpeed$), maximum acceleration ($MaxAcc$) and average length ($AveLength$) measured by a multi-sensor system composed of a land-based radar and an airborne infrared sensor.

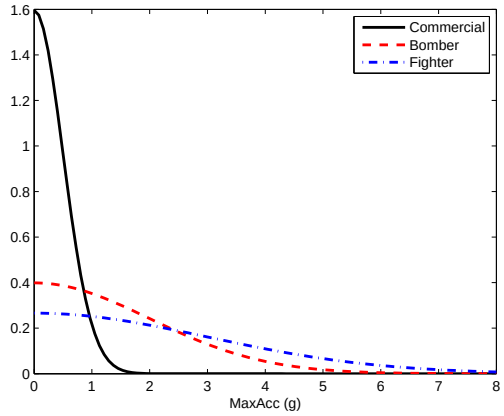
In this numerical study, we simulated the feature measurements using the Gaussian-distributed class-conditional probability functions. We used Gaussian densities, with the parameters selected in such a way that $P\{s_{\min} < x < s_{\max}\} = 0.95$, where $[s_{\min}, s_{\max}]$ is the feature interval given in Table 6.1. Figure 6.3 shows the distributions of the three features conditioned on the class.

Table 6.1: Feature intervals for three airborne target classes

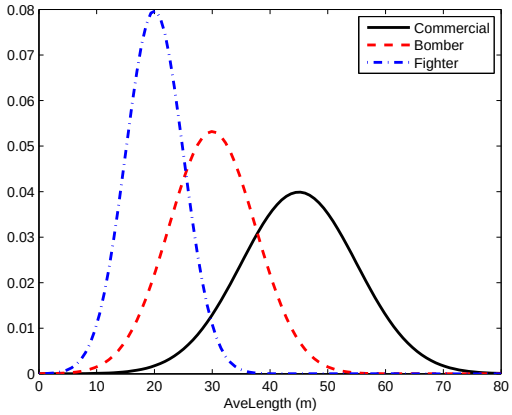
Class	AveSpeed (km/h)	MaxAcc (g)	AveLength (m)
Commercial (ω_1)	[600, 800]	[0, 1]	[25, 65]
Bomber (ω_2)	[400, 700]	[0, 4]	[15, 45]
Fighter (ω_3)	[500, 1000]	[0, 6]	[10, 30]



(a) AveSpeed



(b) MaxAcc



(c) AveLength

Figure 6.3: Distributions of the three features conditioned on the class

6.3.2 Implementation of the hybrid belief rule base

In this study, assume that 120 labeled training samples $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3})$, $i = 1, 2, \cdots, 120$, were collected by the multi-sensor system according to the above class-conditional probability distributions with equal prior probabilities. Further, assume the available training samples are not fully reliable, i.e., some of them have wrong class labels. We simulated this scenario by adding class noise with noise level of $x\%$ indicating $x\%$ of the samples in the training set are mislabeled. The class labels of these samples were randomly changed to different ones within the domain of the class. Apart from the uncertain training data, suppose three pieces of partially reliable expert knowledge $\mathbf{e}_j$, $j = 1, 2, \cdots, 3$, were obtained by structured interview. These two types of information were used to construct the HBRB using the proposed method. The processes of constructing the HBRB concerning the airborne target classification problem were implemented as follows.

Step 1: Preprocess: fuzzification of the feature space

The prerequisite step to generate a BRB is to fuzzify the feature space. We use the fuzzy grid partition method [46] to fuzzify the feature space. Suppose according to the *a priori* knowledge, it is known that the three features *AveSpeed*, *MaxAcc*, and *AveLength*, change in the intervals $[400, 1000]$, $[0, 6]$ and $[10, 70]$, respectively. The partition number for each feature is set to three. As only a few training samples are available, a large partition number may result in over-fitting. Moreover, a relatively small partition number makes it easier for the experts to assign fuzzy regions to each class. Based on the fuzzy grid partition method, the fuzzification of the three features is shown in Figure 6.4.

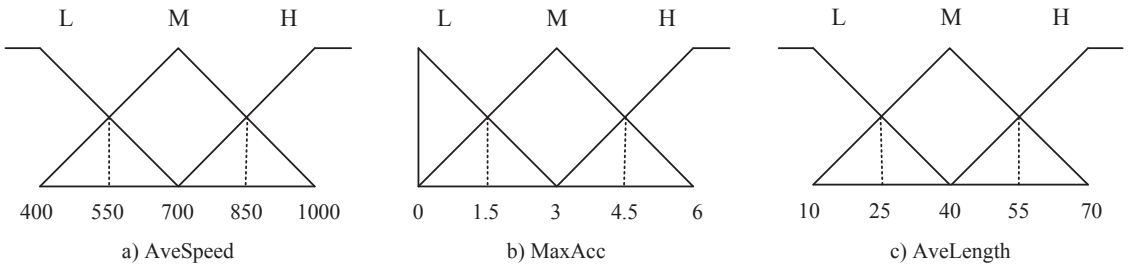


Figure 6.4: Fuzzification of the feature space

Step 2: Construction of DBRB

In this step, 120 labeled training samples $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3})$, $i = 1, 2, \cdots, 120$, were used to construct the DBRB using the method described in Chapter 5. Table 6.2 shows the DBRB containing 12 belief rules learnt from the uncertain training data with class noise level of 30%. Although the DBRB generation method can reduce the adverse effects of class noise, the consequence of one rule may still be unreliable in small training set and excessive noise conditions. For example, in the constructed DBRB, the rule R_D^3 assigns

most belief to ω_3, which is not consistent with the real class-conditional distributions shown in Figure 6.3 (from which, class ω_2 should be assigned more belief). This is because only one training sample is assigned to the corresponding fuzzy region $\{L \wedge M \wedge L\}$, and this only training sample is not representative of the real class-conditional distributions. Fortunately, the developed rule weight generation method only assigns this rule a small weight, which decreases its effect in the reasoning process.

Table 6.2: DBRB constructed based on the uncertain training data

Rule number	Rule weight	Antecedent	Consequent
1	0.45	$L \wedge L \wedge L$	$\{(\omega_1, 0), (\omega_2, 0.3257), (\omega_3, 0.4812)\}$
2	0.69	$L \wedge L \wedge M$	$\{(\omega_1, 0.0041), (\omega_2, 0.8833), (\omega_3, 0.1126)\}$
3	0.36	$L \wedge M \wedge L$	$\{(\omega_1, 0), (\omega_2, 0), (\omega_3, 0.3581)\}$
4	0.56	$L \wedge M \wedge M$	$\{(\omega_1, 0), (\omega_2, 0.6687), (\omega_3, 0)\}$
5	0.90	$M \wedge L \wedge L$	$\{(\omega_1, 0), (\omega_2, 0.1011), (\omega_3, 0.8879)\}$
6	1.00	$M \wedge L \wedge M$	$\{(\omega_1, 0.8632), (\omega_2, 0.1245), (\omega_3, 0)\}$
7	0.54	$M \wedge L \wedge H$	$\{(\omega_1, 0.7906), (\omega_2, 0.1047), (\omega_3, 0.1035)\}$
8	0.44	$M \wedge M \wedge L$	$\{(\omega_1, 0), (\omega_2, 0.7084), (\omega_3, 0.2119)\}$
9	0.69	$M \wedge M \wedge M$	$\{(\omega_1, 0), (\omega_2, 0.8998), (\omega_3, 0)\}$
10	0.63	$H \wedge L \wedge L$	$\{(\omega_1, 0.2007), (\omega_2, 0), (\omega_3, 0.7591)\}$
11	0.43	$H \wedge M \wedge L$	$\{(\omega_1, 0.1171), (\omega_2, 0), (\omega_3, 0.8295)\}$
12	0.36	$H \wedge M \wedge M$	$\{(\omega_1, 0), (\omega_2, 0), (\omega_3, 0.6657)\}$

Step 3: Construction of KBRB

Suppose the following three pieces of expert knowledge $\mathbf{e}_j$, $j = 1, 2, \cdots, 3$, were obtained by asking the experts to assign the fuzzy regions to each target class and to give the corresponding certainty grades:

$\mathbf{e}_1$: If x_1 is $\{M\}$ and x_2 is $\{L\}$ and x_3 is $\{M, H\}$, then consequence is ω_1, with certainty grade 0.9;

$\mathbf{e}_2$: If x_1 is $\{L, M\}$ and x_2 is $\{L, M\}$ and x_3 is $\{L, M\}$, then consequence is ω_2, with certainty grade 0.7;

$\mathbf{e}_3$: If x_1 is $\{M, H\}$ and x_2 is $\{L, M, H\}$ and x_3 is $\{L\}$, then consequence is ω_3, with certainty grade 0.8.

As shown in Table 6.3, each piece of expert knowledge covers several fuzzy regions. We can further see that one fuzzy region $\{M \wedge L \wedge M\}$ is covered by both the expert knowledge $\mathbf{e}_1$ and $\mathbf{e}_2$, and two fuzzy regions $\{M \wedge L \wedge L\}$ and $\{M \wedge M \wedge L\}$ are covered by both the expert knowledge $\mathbf{e}_2$ and $\mathbf{e}_3$. For these three regions, the consequences are obtained by combing the conflicting pieces of expert knowledge considering their certainty grades with Eqs.(6.3-6.5). For those non-overlapping regions, the consequences are kept unchanged. Table 6.4 shows the KBRB containing 13 belief rules constructed from the expert knowledge. In the

constructed KBRB, rule R_K^6 is obtained by combining the conflicting items $\mathbf{e}_1$ and $\mathbf{e}_2$, and rules R_K^5 and R_K^8 are obtained by combining the conflicting items $\mathbf{e}_2$ and $\mathbf{e}_3$. We can see that for the three new fused rules, the consequences are not complete (i.e., the sum of the belief for all of the three classes is less than one), with the left belief characterizing the ignorance induced by the partially reliable expert knowledge. Due to partially available expert knowledge, the generated KBRB only covers part of the fuzzy regions of the feature space. In addition, due to the insufficiency of the expert knowledge, the consequence of some rule may be unreliable. For example, in the constructed KBRB, the rule R_K^1 assigns full belief to ω_2. However, according to the real class-conditional distributions shown in Figure 6.3, the samples from class ω_3 also have a large possibility to fall into the corresponding fuzzy region $\{L \wedge L \wedge L\}$.

Table 6.3: Fuzzy regions covered by each piece of expert knowledge

Expert knowledge	Covered fuzzy regions
$\mathbf{e}_1$	$\{\mathbf{M} \wedge \mathbf{L} \wedge \mathbf{M}\}$, $\{M \wedge L \wedge H\}$
$\mathbf{e}_2$	$\{L \wedge L \wedge L\}$, $\{L \wedge M \wedge L\}$, $\{L \wedge L \wedge M\}$, $\{L \wedge M \wedge M\}$, $\{\mathbf{M} \wedge \mathbf{L} \wedge \mathbf{L}\}$, $\{\mathbf{M} \wedge \mathbf{M} \wedge \mathbf{L}\}$, $\{\mathbf{M} \wedge \mathbf{L} \wedge \mathbf{M}\}$, $\{M \wedge M \wedge M\}$
$\mathbf{e}_3$	$\{\mathbf{M} \wedge \mathbf{L} \wedge \mathbf{L}\}$, $\{\mathbf{M} \wedge \mathbf{M} \wedge \mathbf{L}\}$, $\{M \wedge H \wedge L\}$, $\{H \wedge L \wedge L\}$, $\{H \wedge M \wedge L\}$, $\{H \wedge H \wedge L\}$

Table 6.4: KBRB constructed based on the expert knowledge

Rule number	Rule weight	Antecedent	Consequent
1	0.70	$L \wedge L \wedge L$	$\{(\omega_1, 0), (\omega_2, 1), (\omega_3, 0)\}$
2	0.70	$L \wedge L \wedge M$	$\{(\omega_1, 0), (\omega_2, 1), (\omega_3, 0)\}$
3	0.70	$L \wedge M \wedge L$	$\{(\omega_1, 0), (\omega_2, 1), (\omega_3, 0)\}$
4	0.70	$L \wedge M \wedge M$	$\{(\omega_1, 0), (\omega_2, 1), (\omega_3, 0)\}$
5	1.00	$M \wedge L \wedge L$	$\{(\omega_1, 0), (\omega_2, 0.3182), (\omega_3, 0.5455)\}$
6	1.00	$M \wedge L \wedge M$	$\{(\omega_1, 0.7297), (\omega_2, 0.1892), (\omega_3, 0)\}$
7	0.90	$M \wedge L \wedge H$	$\{(\omega_1, 1), (\omega_2, 0), (\omega_3, 0)\}$
8	1.00	$M \wedge M \wedge L$	$\{(\omega_1, 0), (\omega_2, 0.3182), (\omega_3, 0.5455)\}$
9	0.70	$M \wedge M \wedge M$	$\{(\omega_1, 0), (\omega_2, 1), (\omega_3, 0)\}$
10	0.80	$M \wedge H \wedge L$	$\{(\omega_1, 0), (\omega_2, 0), (\omega_3, 1)\}$
11	0.80	$H \wedge L \wedge L$	$\{(\omega_1, 0), (\omega_2, 0), (\omega_3, 1)\}$
12	0.80	$H \wedge M \wedge L$	$\{(\omega_1, 0), (\omega_2, 0), (\omega_3, 1)\}$
13	0.80	$H \wedge H \wedge L$	$\{(\omega_1, 0), (\omega_2, 0), (\omega_3, 1)\}$

Step 4: Construction of HBRB

In this stage, the rules from the DBRB (Table 6.2) and the KBRB (Table 6.4) are combined based on the fusion algorithm developed in Section 6.2.2. Table 6.5 shows the optimal HBRB containing 14 belief rules by optimizing the average leave-one-out test

error. Compared to the previous generated DBRB and KBRB, the integrated HBRB has the following two main advantages:

1. It covers more fuzzy regions than both the DBRB and the KBRB, so that it is more powerful to classify those patterns uncovered by either the DBRB or the KBRB.

2. In the overlapping fuzzy regions of the DBRB and the KBRB, through combination, the rules in HBRB reduced the potential unreliability existing in the corresponding rules of DBRB or KBRB. For example, as indicated in *Step 2*, the rule R_D^3 generated from the uncertain training data is unreliable, but after combination with the corresponding rule R_K^3, the consequence of the combined rule R_H^3 has better representation for the real class distributions in the fuzzy region $\{L \wedge M \wedge L\}$. Similarly, as indicated in *Step 3*, the rule R_K^1 generated from the uncertain expert knowledge is unreliable, but after combination with the corresponding rule R_D^1, a better rule R_H^1 is generated for the fuzzy region $\{L \wedge L \wedge L\}$.

Table 6.5: HBRB constructed based on the uncertain training data and expert knowledge

Rule number	Rule weight	Antecedent	Consequent
1 (R_D^1,R_K^1)[a]	0.52	$L \wedge L \wedge L$	$\{(\omega_1, 0), (\omega_2, 0.3853), (\omega_3, 0.2833)\}$
2 (R_D^2,R_K^2)	0.69	$L \wedge L \wedge M$	$\{(\omega_1, 0.0022), (\omega_2, 0.7346), (\omega_3, 0.0614)\}$
3 (R_D^3,R_K^3)	0.45	$L \wedge M \wedge L$	$\{(\omega_1, 0), (\omega_2, 0.2146), (\omega_3, 0.2053)\}$
4 (R_D^4,R_K^4)	0.60	$L \wedge M \wedge M$	$\{(\omega_1, 0), (\omega_2, 0.6263), (\omega_3, 0)\}$
5 (R_D^5,R_K^5)	0.93	$M \wedge L \wedge L$	$\{(\omega_1, 0), (\omega_2, 0.0930), (\omega_3, 0.6786)\}$
6 (R_D^6,R_K^6)	1.00	$M \wedge L \wedge M$	$\{(\omega_1, 0.6873), (\omega_2, 0.0918), (\omega_3, 0)\}$
7 (R_D^7,R_K^7)	0.63	$M \wedge L \wedge H$	$\{(\omega_1, 0.7976), (\omega_2, 0.0025), (\omega_3, 0.0018)\}$
8 (R_D^8,R_K^8)	0.59	$M \wedge M \wedge L$	$\{(\omega_1, 0), (\omega_2, 0.5153), (\omega_3, 0.2084)\}$
9 (R_D^9,R_K^9)	0.69	$M \wedge M \wedge M$	$\{(\omega_1, 0), (\omega_2, 0.8028), (\omega_3, 0)\}$
10 ($-$,R_K^{10})	0.22	$M \wedge H \wedge L$	$\{(\omega_1, 0), (\omega_2, 0), (\omega_3, 1)\}$
11 (R_D^{10},R_K^{11})	0.68	$H \wedge L \wedge L$	$\{(\omega_1, 0.0004), (\omega_2, 0), (\omega_3, 0.8024)\}$
12 (R_D^{11},R_K^{12})	0.53	$H \wedge M \wedge L$	$\{(\omega_1, 0.6039), (\omega_2, 0), (\omega_3, 0.7052)\}$
13 (R_D^{12},$-$)	0.26	$H \wedge M \wedge M$	$\{(\omega_1, 0), (\omega_2, 0), (\omega_3, 0.6657)\}$
14 ($-$,R_K^{13})	0.22	$H \wedge H \wedge L$	$\{(\omega_1, 0), (\omega_2, 0), (\omega_3, 1)\}$

[a]The corresponding rules in DBRB and KBRB with the same antecedent parts are shown in brackets.

6.3.3 Comparative study

The HBRBCS was compared to the DBRBCS ($\lambda = 1$, which only considers the uncertain training data) and the KBRBCS ($\lambda = 0$, which only considers the uncertain expert knowledge) under different noise levels for the training data. A test set of 3000 samples

drawn from the original class-conditional probability distributions were used for error estimation. For the HBRBCS, the optimal adjustment factor λ by optimizing the average leave-one-out test error was used to get the integrated HBRB. In addition, two well-known robust data-based classifiers, C4.5 [83] and FRBCS [17], as well as a representative hybrid classifier AFRBCS [107] reviewed in Section 2.4.3 were also considered in the comparison.

Table 6.6 shows the classification error rates for different methods with different noise levels. With the increase of the class noise in the training data set, the performance of all of the three data-based classifiers C4.5, FRBCS, and DBRBCS, decrease, whereas the DBRBCS shows more robust to class noise due to the utilized belief rule structure and belief reasoning method. The KBRBCS, which classifies query patterns only based on the expert knowledge, always yields a moderate performance. Interestingly, the HBRBCS outperforms both the DBRBCS and the KBRBCS with any noise level. The reason is that, on the one hand, the fused HBRB covers more fuzzy regions than do the partial DBRB and KBRB and, on the other hand, for the overlapping fuzzy regions, thanks to combination, the rules in HBRB reduced the potential unreliability existing in the corresponding rules of DBRB or KBRB. Besides, by comparing the two hybrid classifiers AFRBCS and HBRBCS, it can be seen that the proposed HBRBCS yields better performance, especially for cases with high data noise levels. It is because in HBRBCS the weights of training data and expert knowledge are adjusted adaptively according to the qualities of these two types of information. Whereas, the AFRBCS always uses training data to update the knowledge-based model, no matter the available training data set is reliable or not.

Table 6.6: Classification error rates (in %) for considered methods with different noise levels

NL	C4.5	FRBCS	DBRBCS	KBRBCS	AFRBCS	HBRBCS	CI[a]
0%	19.30	18.13	17.87	23.37	18.23	**17.30**[b]	[15.97,18.71]
10%	20.60	21.33	18.83	23.37	20.10	**17.83**	[16.48,19.24]
20%	23.87	24.37	20.47	23.37	22.20	**18.47**	[17.10,19.92]
30%	26.63	28.43	24.03	23.37	23.93	**18.83**	[17.44,20.28]
40%	34.77	35.60	30.70	23.37	26.67	**19.47**	[18.07,20.92]
50%	40.40	41.57	38.13	23.37	30.47	**20.40**	[18.97,21.91]

[a] The last column is the 95% confidence interval of the best method.

[b] Results in boldface correspond to the lowest error rate.

In order to find out whether significant differences exist among different methods, error estimation confidence intervals (with confidence level $A = 95\%$) [1] for the best method

[1] Computed by numerically solving the equations: $\sum_{k \geq K} P(k, N, \underline{p}) = (1 - A)/2$ and $\sum_{k \leq K} P(k, N, \overline{p}) = (1 - A)/2$, where $P(k, N, p)$ is the binomial distribution, N is the number of test patterns, K is the number of patterns misclassified, A is the confidence level, and $[\underline{p}, \overline{p}]$ is the confidence interval.

(i.e., HBRBCS) are shown in the last column. It is interesting to note that only when the noise level is quite low (0% and 10%), the error rate of the second best method is within the corresponding confidence interval. When the class noise increases, the best HBRBCS method shows a statistically significant advantage. Therefore, the classification performance can improve greatly by making use of the complementary information from uncertain training data and expert knowledge based on belief function theory, especially when both sources of information have high uncertainty.

6.3.4 Parameter analysis

The adjustment factor λ plays an important role in determining the classification performance of the HBRBCS. In this section, we take an analysis for the effect of the adjustment factor λ and evaluate whether the optimization method developed in Section 6.2.2 works well.

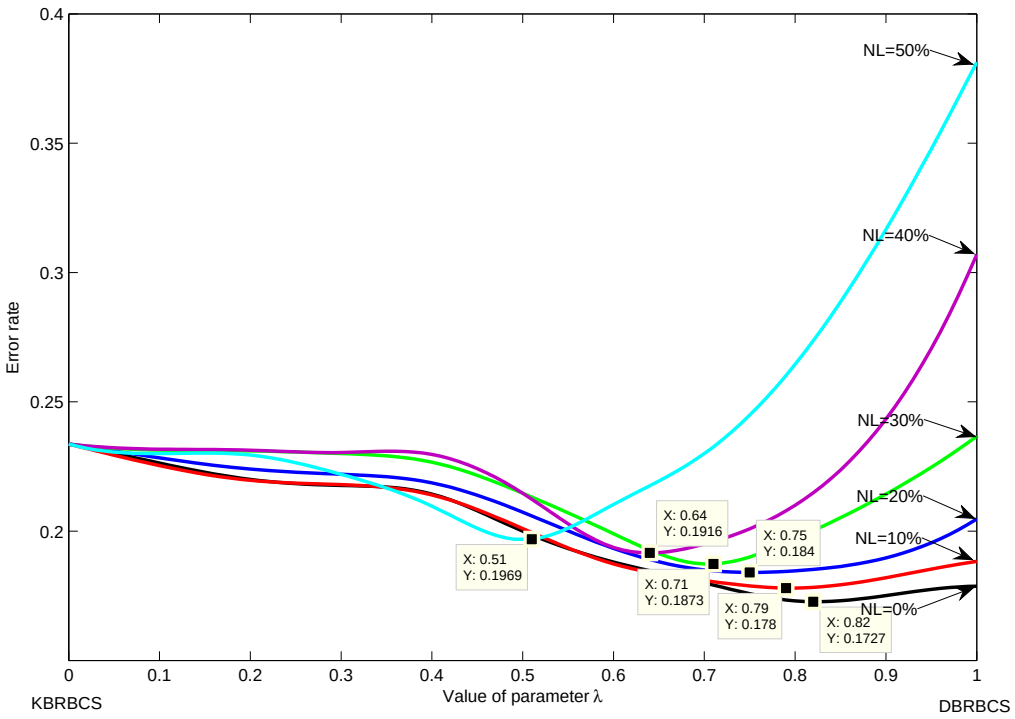


Figure 6.5: Classification error rate of the HBRBCS with the adjustment factor ranging from 0 to 1

Figure 6.5 shows the classification error rate of the HBRBCS with the adjustment factor ranging from 0 to 1 under different noise levels. It can be seen that the optimal values of the adjustment factor λ are different under different noise levels. With the increase of the noise level, the optimal value of λ tends to be smaller. Because in these cases, the DBRB generated from the noisy training data becomes less reliable, and the KBRB which is not affected by the noisy training data takes more important role in determining the

classification.

Table 6.7 compares the estimated value of λ by optimizing the average leave-one-out test error with the tested optimal value of λ indicated in Figure 6.5. It can be seen that the parameter optimization method provides a good estimation for the optimal value of λ. Accordingly, the classification error rate with the estimated optimal value of λ is very close to the tested optimal error rate. In addition, the estimation is more accurate with low noise levels, because in these cases the training data has a good representation of the real class-conditional distributions.

Table 6.7: Comparison of the estimated and tested optimal λ as well as their corresponding classification error rates (in %) with different noise levels

Noise Level	0%	10%	20%	30%	40%	50%
Est. λ	0.80	0.77	0.78	0.73	0.68	0.56
Tes. λ	0.82	0.79	0.75	0.71	0.64	0.51
Error rate with Est. λ	17.30	17.83	18.47	18.83	19.47	20.40
Error rate with Tes. λ	17.27	17.80	18.40	18.73	19.16	19.69

6.4 Conclusion

In order to make use of the information from both uncertain training data and expert knowledge for classification, a hybrid belief rule-based classification system (HBRBCS) has been developed based on the belief rule structure. The proposed HBRBCS offers complementary advantages from data-driven models and knowledge-driven models. This system can be useful for many real-world applications where both uncertain training data and expert knowledge are available. An airborne target classification in the air surveillance system has been studied to present the implementation of the proposed HBRBCS and to demonstrate its capacity of combining both uncertain training data and expert knowledge for classification. The experiment results show that the HBRBCS can make good use of these two types of independent and complementary information and achieve better performance.

Conclusions and future research directions

Conclusions

This thesis has tackled uncertain data classification problems in nearest-neighbor-based and rule-based approaches, by introducing belief functions to model the uncertainty in training data or expert knowledge. The following four main contributions were exposed in this thesis.

First, in order to model the imprecise information in class overlapping regions, an evidential editing procedure was designed to reassign the original training samples with new labels represented by an evidential membership structure. Based on the evidential editing procedure, we have developed an evidential editing version of the k-nearest neighbor rule (EEkNN). Experiments have shown that the proposed EEkNN classifier can achieve better performance than other considered nearest-neighbor-based methods, especially for data sets with high overlapping ratios. In addition, the proposed EEkNN classifier is not too sensitive to the value of k, which is very useful for those time or space-critical applications in which only a small value of k is permitted in the classification process.

Second, in order to improve the performance of the kNN-based classifier based on incomplete training data set, we have designed an evidential fusion scheme for combining a group of pairwise kNN classifiers in the framework of belief functions (PkNN-BF). Each pairwise kNN classifier was locally learned based on a pairwise distance metric, which provides greater flexibility to design the feature weights so that the local specificities in feature space can be well characterized. From the reported experiment results, we can conclude that the proposed PkNN-BF classifier can successfully improve the performance for those tasks with high dimension and small sample size, in which cases the training data set is not rich enough to well characterize the real class-conditional probability distributions.

Third, we have extended the traditional fuzzy rule-based classification system in the framework of belief functions and developed a belief rule-based classification system (BRBCS) to address uncertain information in complex classification problems. The two components of the proposed BRBCS, i.e., the belief rule base and the belief reasoning method,

have been designed specifically by taking into account the possible pattern noise in many real-world data sets. The delivered experiments have shown that the proposed BRBCS can get better classification accuracy and robustness than other rule-based methods for a variety of real-world classification problems. This allows us to conclude that the introduction of belief functions have improved the behavior of the rule-based classification system.

Fourth, a hybrid belief rule-based classification system (HBRBCS) has been developed based on the belief rule structure in order to make use of the information from both uncertain training data and expert knowledge jointly for classification. The proposed HBRBCS can inherit the complementary advantages from data-driven models and knowledge-driven models, and so it is quite useful for many real-world applications where both uncertain training data and expert knowledge are available. We have studied an airborne target classification problem, in which both training data collected by sensors and expert knowledge are available. The experiment results have shown that the HBRBCS can make good use of these two types of independent and complementary information and achieve better performance.

Future research directions

The work presented in this thesis can be continued in many directions. In the following paragraphs, we sketch a few of them.

For short-term perspectives, some of the methods developed in this thesis can be further extended. Our first concern is the distance metric leaning problem in kNN rule. In Chapter 4, we have proposed a pairwise distance metric related to pairs of class labels. However, as reviewed in Section 2.3.3, there is another way to learn local distance metric, which is based on the geometric location of patterns. Both the geometric location and class label are important local information for the available patterns. Therefore, if we further consider the location information of patterns in our proposed pairwise distance metric, the behavior of kNN rule may be further improved. Another concern is the membership function in rule-based classification systems. For our proposed belief rule-based classification system in Chapter 5, we have used the traditional triangular membership function for the antecedent fuzzy partitions of each rule. As reviewed in Section 2.4.2, recently, there were some proposals to learn optimized membership functions with genetic algorithms. These optimized membership functions may further improve the performance of our proposed belief rule-based classification system.

In the long term, we will focus on some other important uncertain data classification problems. One such challenging problem emerging in machine learning field is sequence labeling. For the classification problems considered in this thesis, the samples are assumed to be collected independently. However, for some real-world applications, such as speech

recognition and handwriting recognition, the samples are collected sequentially and there exist potential relations between nearby observations. It is expected that these uncertain constraints are helpful for classification. We plan to build hidden Markov models in the framework of belief functions to model the uncertain relationship for the labels of nearby observations.

Bibliography

[1] C. C. Aggarwal and P. S. Yu. A survey of uncertain data algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, 21:609–623, 2009.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, San Francisco, USA, 1994.

[3] M. R. Akbarzadeh-Totonchi and M. Moshtagh-Khorasani. A hierarchical fuzzy rule-based approach to aphasia diagnosis. *Journal of Biomedical Informatics*, 40:465–475, 2007.

[4] J. Alcala-Fdez, R. Alcala, and F. Herrera. A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. *IEEE Transactions on Fuzzy Systems*, 19:857–872, 2011.

[5] J. M. Alonso and L. Magdalena. Special issue on interpretable fuzzy systems. *Information Sciences*, 181:4331–4339, 2011.

[6] F. Angiulli. Fast condensed nearest neighbor rule. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 7–11, New York, USA, 2005.

[7] A. Appriou. Uncertain data aggregation in classification and tracking processes. In B. Bouchon-Meunier, editor, *Aggregation and Fusion of Imperfect Information*, volume 12, pages 231–260. Heidelberg, Germany: Physica-Verlag, 1998.

[8] M. Avriel. *Nonlinear Programming: Analysis and Methods*. Chelmsford, MA: Courier Corporation, 2003.

[9] A. M. Aziz. A new multiple decisions fusion rule for targets detection in multiple sensors distributed detection systems with data fusion. *Information Fusion*, 18:175–186, 2014.

[10] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *Proceedings of 20th International Conference on Machine Learning*, pages 11–18, Washington D.C., USA, 2003.

[11] J. A. Barnett. Calculating Dempster-Shafer plausibility. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:599–602, 1991.

[12] E. Binaghi and P. Madella. Fuzzy Dempster-Shafer reasoning for rule-based classifiers. *International Journal of Intelligent Systems*, 14:559–583, 1999.

[13] R. H. Bonczek, C. W. Holsapple, and A. B. Whinston. *Foundations of Decision Support Systems*. London: Academic Press, 2014.

[14] B. G. Buchanan and E. H. Shortliffe. *Rule-based Expert Systems*. Reading, MA: Addison-Wesley, 1984.

[15] L. H. Chen. An extended rule-based inference for general decision-making problems. *Information Sciences*, 102:247–261, 1997.

[16] Y. Chen and J. Z. Wang. Support vector learning for fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 11:716–728, 2003.

[17] Z. Chi, H. Yan, and T. Pham. *Fuzzy Algorithms with Applications to Image Processing and Pattern Recognition*. Singapore: World Scientific, 1996.

[18] O. Cordon, F. Herrera, F. Hoffman, and L. Magdalena. *Genetic Fuzzy Systems*. Singapore: World Scientific, 2001.

[19] O. Cordón, M. José del Jesus, and F. Herrera. A proposal on reasoning methods in fuzzy rule-based classification systmes. *International Journal of Approximate Reasoning*, 20:21–45, 1999.

[20] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.

[21] S. Das. *High-Level Data Fusion*. Boston, MA: Artech House, 2008.

[22] B. Dasarathy. Noising around the neighbourhood: a new system structure and classification rule for recognition in partially exposed environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:67–71, 1980.

[23] P. Dasgupta. A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 38:549–563, 2008.

[24] A. Dempster. Upper and lower probabilities induced by multivalued mapping. *Annals of Mathematical Statistics*, 38:325–339, 1967.

[25] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[26] T. Denœux. A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics*, 25:804–813, 1995.

[27] T. Denœux. Analysis of evidence-theoretic decision rules for pattern classification. *Pattern Recognition*, 30:1095–1107, 1997.

[28] T. Denœux. A neural network classifier based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 30:131–150, 2000.

[29] T. Denœux. Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artificial Intelligence*, 172:234–264, 2008.

[30] T. Denœux. *Introduction to Belief Functions*. Lecture notes, Université de Technologie de Compiègne, Compiègne, France, 2011.

[31] T. Denœux and P. Smets. Classification using belief functions: the relationship between the case-based and model-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36:1395–1406, 2006.

[32] T. Denœux and A. B. Yaghlane. Approximating the combination of belief functions using the fast Möbius transform in a coarsened frame. *International Journal of Approximate Reasoning*, 31:77–101, 2002.

[33] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Berlin: Springer-Verlag, 1996.

[34] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.

[35] D. Dubois and H. Prade. A set-theoretic view of belief functions: logical operations and approximations by fuzzy sets. *International Journal of General System*, 12:193–226, 1986.

[36] S. A. Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, 6:325–327, 1972.

[37] C. F. Eick, A. Rouhana, A. Bagherjeiran, and R. Vilalta. Using clustering to learn distance functions for supervised similarity assessment. *Engineering Applications of Artificial Intelligence*, 19:395–401, 2006.

[38] E. Fix and J. Hodges. Discriminatory analysis, nonparametric discrimination: consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[39] J. H. Friedman. Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University, Stanford, California, 1996.

[40] S. I. Gallant. Perceptron-based learning algorithms. *IEEE Transactions on Neural Networks*, 1:179–191, 1990.

[41] S. García, A. Fernández, J. Luengo, and F. Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180:2044–2064, 2010.

[42] E. Haenni. Are alternatives to Dempster's rule of combination real alternatives. *Information Fusion*, 3:237–239, 2002.

[43] H. Helbig. *Knowledge Representation and the Semantics of Natural Language.* New York, NY: Springer, 2006.

[44] C. W. Holsapple, A. B. Whinston, J. H. Benamati, and G. S. Kearns. *Decision Support Systems: A Knowledge-Based Approach.* Saint Paul, MN: West Publishing, 1996.

[45] J. Hühn and E. Hüllermeier. FURIA: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, 19:293–319, 2009.

[46] H. Ishibuchi, K. Nozaki, and H. Tanaka. Distributed representation of fuzzy rules and its application to pattern classification. *Fuzzy Sets and Systems*, 52:21–32, 1992.

[47] H. Ishibuchi, T. Yamamoto, and T. Nakashima. Hybridization of fuzzy GBML approaches for pattern classification problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35:359–365, 2005.

[48] P. Jackson. *Introduction to Expert Systems.* Boston, MA: Addison-Wesley, 1990.

[49] M. Z. Jahromi, E. Parvinnia, and R. John. A method of learning weighted similarity function to improve the performance of nearest neighbor. *Information Sciences*, 179:2964–2973, 2009.

[50] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:4–37, 2000.

[51] F. V. Jensen. *An Introduction to Bayesian Networks.* London: UCL press, 1996.

[52] A. Jøsang. The consensus operator for combining beliefs. *Artificial Intelligence*, 141:157–170, 2002.

[53] A. L. Jousselme, D. Grenier, and E. Bossé. A new distance between two bodies of evidence. *Information Fusion*, 2:91–101, 2001.

[54] A. L. Jousselme and P. Maupin. Distances in evidence theory: Comprehensive survey and generalizations. *International Journal of Approximate Reasoning*, 53:118–145, 2012.

[55] J. M. Keller, M. R. Gray, and J. A. Givens. A fuzzy k-nearest neighbor algorithm. *IEEE Transactions on Systems, Man and Cybernetics*, 15:580–585, 1985.

[56] E. P. Klement, R. Mesiar, and E. Pap. *Triangular Norms.* Dordrecht: Springer, 2000.

[57] G. L. Kong, D. L. Xu, and Richard B. A belief rule-based decision support system for clinical risk assessment of cardiac chest pain. *European Journal of Operational Research*, 219:564–573, 2012.

[58] J. Koplowitz and T. A. Brown. On the relation of performance to editing in nearest neighbor rules. *Pattern Recognition*, 13:251–255, 1981.

[59] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. Machine learning: A review of classification and combining techniques. *Artificial Intelligence Review*, 26:159–190, 2006.

[60] L. Kuncheva, J. Bezdek, and R. Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34:299–314, 2001.

[61] H. Laanaya, A. Martin, D. Aboutajdine, and A. Khenchaf. Support vector regression of membership functions and belief functions - application for pattern recognition. *Information Fusion*, 11:338–350, 2010.

[62] E. Lefevre, O. Colot, and P. Vannoorenberghe. Belief functions combination and conflict management. *Information Fusion*, 3:149–162, 2002.

[63] B. Li, H. W. Wang, J. B. Yang, M. Guo, and C. Qi. A belief-rule-based inventory control method under nonstationary and uncertain demand. *Expert Systems with Applications*, 38:14997–15008, 2011.

[64] J. Liu, L. Martinez, A. Calzada, and H. Wang. A novel belief rule base representation, generation and its inference methodology. *Knowledge-Based Systems*, 53:129–141, 2013.

[65] J. Liu, J. B. Yang, J. Wang, H. S. Sii, and Y. M. Wang. Fuzzy rule based evidential reasoning approach for safety analysis. *International Journal of General Systems*, 33:183–204, 2004.

[66] W. Liu. Analyzing the degree of conflict among belief functions. *Artificial Intelligence*, 170:909–924, 2006.

[67] Z. Liu. *Credal classification of uncertain data based on belief function theory.* PhD thesis, Université de Bretagne Occidentale, Brest, France, 2014.

[68] Z. Liu, Q. Pan, and J. Dezert. A new belief-based k-nearest neighbor classification method. *Pattern Recognition*, 46:834–844, 2013.

[69] Z. Liu, Q. Pan, J. Dezert, and G. Mercier. Credal classification rule for uncertain data based on belief functions. *Pattern Recognition*, 44:2532–2541, 2014.

[70] B. Markman. *Knowledge Representation.* London: Lawrence Erlbaum Associates, 1999.

[71] K. L. McGraw and B. K. Harbison-Briggs. *Knowledge Acquisition, Principles and Guidelines.* Upper Saddle River, NJ: Prentice Hall, 1989.

[72] C. J. Merz, P. M. Murphy, and D. W. Aha. UCI repository of machine learning databases. Technical report, Deptartment of Information and Computer Science, University of California, Irvine, California, 1997. http://www.ics.uci.edu/mlearn/MLRepository.html.

[73] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine Learning, Neural and Statistical Classification*. New York, NY: Ellis Horwood, 1994.

[74] J. Montero and D. Ruan. Modelling uncertainty (editorial). *Information Sciences*, 180:799–802, 2010.

[75] R. L. Morin and D. E. Raeside. A reappraisal of distance-weighted k-nearest-neighbor classification for pattem recognition with missing data. *IEEE Transactions on Systems, Man, and Cybernetics*, 11:241–243, 1981.

[76] L. Nanni and A. Lumini. Prototype reduction techniques: A comparison among different approaches. *Expert Systems with Applications*, 38:11820–11828, 2011.

[77] H. T. Nguyen. *An Introduction to Random Sets*. London: Chapman and Hall, 2006.

[78] K. Nozaki, H. Ishibuchi, and H. Tanaka. Adaptive fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 4:238–250, 1996.

[79] R. Paredes and E. Vidal. A class-dependent weighted dissimilarity measure for nearest neighbor classication problems. *Pattern Recognition Letters*, 21:1027–1036, 2000.

[80] R. Paredes and E. Vidal. Learning weighted metrics to minimize nearest-neighbor classification error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1100–1110, 2006.

[81] B. Qin, Y. Xia, S. Prabhakar, and Y. Tu. A rule-based classification algorithm for uncertain data. In *Proceedings of the 25th IEEE International Conference on Data Engineering*, pages 1633–1640, Shanghai, China, 2009.

[82] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[83] J. R. Quinlan. *C4.5: Programs for Machine Learning*. San Francisco, CA: Morgan Kaufmann Publishers, 1993.

[84] A. Quteishat, C. P. Lim, and S. T. Kay. A modified fuzzy mińcmax neural network with a genetic-algorithm-based rule extractor for pattern classification. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40:641–650, 2010.

[85] S. Raudys and A. Jain. Small sample effects in statistical pattern recognition: recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:252–264, 1991.

[86] J. Ren, S. D. Lee, X. Chen, B. Kao, R. Cheng, and D. Cheung. Naive Bayes classification of uncertain data. In *Proceedings of the 25th IEEE International Conference on Data Engineering*, pages 944–949, Shanghai, China, 2009.

[87] B. Ristic and P. Smets. Target classification approach based on the belief function theory. *IEEE Transactions on Aerospace and Electronic Systems*, 41:574–583, 2005.

[88] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65:386–408, 1958.

[89] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362. Cambridge, MA: MIT Press, 1986.

[90] M. Rychener. *Expert Systems for Engineering Design*. London: Academic Press, 2012.

[91] J. A. Sáez, M. Galar, J. Luengo, and F. Herrera. Tackling the problem of classification with noisy data using Multiple Classifier Systems: Analysis of the performance and robustness. *Information Sciences*, 247:1–20, 2013.

[92] J. A. Sáez, J. Luengo, and F. Herrera. On the suitability of fuzzy rule-based classification systems with noisy data. *IEEE Transactions on Fuzzy Systems*, 20:1–19, 2012.

[93] S. R. Samantaray. Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. *Applied Soft Computing*, 13:928–938, 2013.

[94] J. Sanz, A. Fernández, H. Bustince, and F Herrera. A genetic tuning to improve the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets: degree of ignorance and lateral position. *International Journal of Approximate Reasoning*, 52:751–766, 2011.

[95] J. A. Sanz, A. Fernández, H. Bustince, and F. Herrera. Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning. *Information Sciences*, 180:3674–3685, 2010.

[96] G. Shafer. *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton University Press, 1976.

[97] F. Smarandache and J. Dezert. *Advances and Applications of DSmT for Information Fusion*, volume 1-3. Rehoboth, MA: American Research Press, 2004-2009.

[98] P. Smets. The degree of belief in a fuzzy event. *Information sciences*, 25:1–19, 1981.

[99] P. Smets. The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:447–458, 1990.

[100] P. Smets. Varieties of ignorance and the need for well-founded theories. *Information Sciences*, 57:135–144, 1991.

[101] P. Smets. Belief functions: The disjunctive rule of combination and the generalized Bayesian theorem. *International Journal of Approximate Reasoning*, 9:1–35, 1993.

[102] P. Smets. Analyzing the combination of conflicting belief functions. *Information Fusion*, 8:387–412, 2007.

[103] P. Smets and R. Kennes. The transferable belief model. *Artificial intelligence*, 66:191–234, 1994.

[104] D. G. Stavrakoudis, G. N. Galidaki, I. Z. Gitas, and J. B. Theocharis. A genetic fuzzy-rule-based classifier for land cover classification from hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 50:130–148, 2012.

[105] C. Y. Suen, Y. S. Huang, and K. Liu. The combination of multiple classifiers by a neural network approach. *International Journal of Pattern Recognition and Artificial Intelligence*, 9:579–597, 1995.

[106] D. W. Tang, J. B. Yang, and K. S. Chin. A methodology to generate a belief rule base for customer perception risk analysis in new product development. *Expert Systems with Applications*, 38:5373–5383, 2011.

[107] W. Tang, K. Z. Mao, L. O. Mak, and G. W. Ng. Adaptive fuzzy rule-based classification system integrating both expert knowledge and data. In *Proceedings of IEEE 24th International Conference on Tools with Artificial Intelligence*, pages 814–821, Athens, Greece, 2012.

[108] W. Tang, K. Z. Mao, L. O. Mak, G. W. Ng, Z. Sun, J. H. Ang, and G. Lim. Target classification using knowledge-based probabilistic model. In *Proceedings of the 14th International Conference on Information Fusion*, pages 1–8, Chicago, USA, 2011.

[109] G. R. Terrell and D. W. Scott. Variable kernel density estimation. *The Annals of Statistics*, 20:1236–1265, 1992.

[110] I. Tomek. An experiment with the edited nearest neighbor. *IEEE Transactions on Systems, Man, and Cybernetics*, 6:121–126, 1976.

[111] C. Tsang, S. Kwong, and H. Wang. A systematic fuzzy rule based approach for fault classification in transmission lines. *Pattern Recognition*, 40:2373–2391, 2007.

[112] S. Tsang, B. Kao, K. Y. Yip, W. S. Ho, and S. D. Lee. Decision trees for uncertain data. *IEEE Transactions on Knowledge and Data Engineering*, 23:64–78, 2011.

[113] P. Vannoorenbergue and T. Denoeux. Handling uncertain labels in multiclass problems using belief decision trees. In *Proceedings of the 9th International Conference on Information*

Processing and Management of Uncertainty in Knowledge-Based Systems, pages 1919–1926, Annecy, France, 2002.

[114] V. Vapnik. *The Nature of Statistical Learning Theory*. New York, NY: Springer, 1995.

[115] F. Vazquez, J. S. Sanchez, and F. Pla. A stochastic approach to Wilson's editing algorithm. In *Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis*, pages 35–42, Estoril, Portugal, 2005.

[116] S. A. Vinterbo, E.-Y. Kim, and L. Ohno-Machado. Small, fuzzy and interpretable gene expression based classifiers. *Bioinformatics*, 21:1964–1970, 2005.

[117] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. London: Chapman and Hall, 1990.

[118] J. Wang, P. Neskovic, and L. N. Cooper. Improving nearest neighbor rule with a simple adaptive distance measure. *Pattern Recognition Letters*, 28:207–213, 2007.

[119] S. Wang and R. Jin. An information geometry approach for distance metric learning. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, pages 591–598, Clearwater, USA, 2009.

[120] A. R. Webb. *Statistical Pattern Recognition*. Chichester: John Wiley & Sons, 2003.

[121] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[122] D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data sets. *IEEE Transactions on Systems, Man and Cybernetics*, 2:408–421, 1972.

[123] N Wilson. Algorithms for Dempster-Shafer theory. In D. M. Gabbay and P. Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management*, volume 5, pages 421–475. Boston, MA: Kluwer, 2000.

[124] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. *Advances in Neural Information Processing Systems*, 15:521–528, 2003.

[125] P. Xu. *Information Fusion for Scene Understanding*. PhD thesis, Université de Technologie de Compiène, Compiène, France, 2014.

[126] R. Yager. On the Dempster-Shafer framework and new combination rules. *Information Sciences*, 41:93–138, 1987.

[127] R. R. Yager and D. P. Filev. Including probabilistic uncertainty in fuzzy logic controller modeling using Dempster-Shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics*, 25:1221–1230, 1995.

[128] J. B. Yang, J. Liu, J. Wang, H. S. Sii, and H. W. Wang. Belief rule-based inference methodology using the evidential reasoning approach–RIMER. *IEEE Transactions on Systems Man and Cybernetics, Part A: Systems and Humans*, 36:266–285, 2006.

[129] J. B. Yang, I. M. Wang, D. L. Xu, K. S. Chin, and L. Chatton. Belief rule-based methodology for mapping consumer preferences and setting product targets. *Expert Systems with Applications*, 39:4749–4759, 2012.

[130] L. Yang and R. Jin. Distance metric learning: a comprehensive survey. Technical report, Department of Computer Science and Engineering, Michigan State University, East Lansing, Michigan, 2006.

[131] Y. Ying and P. Li. Distance metric learning with eigenvalue optimization. *Journal of Machine Learning Research*, 13:1–26, 2012.

[132] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[133] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

[134] Z. J. Zhou, C. H. Hu, J. B. Yang, D. L. Xu, M. Y. Chen, and D. H. Zhou. A sequential learning algorithm for online constructing belief-rule-based systems. *Expert System with Applications*, 37:1790–1799, 2010.

[135] Z. J. Zhou, C. H. Hu, J. B. Yang, D. L. Xu, and D. H. Zhou. Online updating belief rule based system for pipeline leak detection under expert intervention. *Expert Systems with Applications*, 36:7700–7709, 2009.

[136] X. Zhu and X. Wu. Class noise vs. attribute noise: a quantitative study. *Artificial Intelligence Review*, 22:177–210, 2004.

[137] L. M. Zouhal and T. Denœux. An evidence-theoretic k-NN rule with parameter optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 28:263–271, 1998.