Multi-label learning under probability, veristic variable and belief function theories

Zoulficar Younes

UMR CNRS 6599 - Heudiasyc

Université de Technologie de Compiègne

A thesis submitted for the degree of

PhilosophiæDoctor (PhD)

December 2010

Contents

List of Figures

List of Tables

Introduction

Multi-label learning deals with the problems where each instance can belong to multiple classes at once. It has found many real world applications, such as text categorization and semantic scene classification. In such problems, the learning task consists in predicting a set of labels for each new instance, based on a training set. Traditional single-label learning tasks (binary or multi-class classification) are a special case of multi-label learning task, where each instance is assigned only one class. In single-label learning, all possible classes are considered to be mutually exclusive. In contrast, in multi-label learning, the classes are not necessary exclusive, and they are usually correlated in the sense that, the assignment of an instance to a certain class may provide information about the membership of this instance to other classes. It is known that taking label correlation into account is a crucial issue in multi-label learning and it may improve the classification performance. Multi-label learning problems are thus more difficult to solve than single-label ones.

The most widely used approaches transform a multi-label classification problem into multiple independent binary classification problems, and thus use any conventional classifier for this purpose. The transformation usually follows one-vs-all referred here to as Binary Relevance approach, i.e. a binary classifier for each possible class and the outputs of all classifiers are combined for final decision, or one-vs-one referred here to as Label Ranking approach, i.e. a binary classifier for each pair of classes, the classes are then ranked according to the number of received votes, and are finally split into relevant and non-relevant classes by thresholding. The main limitation of these approaches is that they usually fail to capture the correlation between classes. Another approach referred to as Label Powerset considers each label combination appearing in the training set as a separate class, and thus, it transforms the multi-label classification

problem into a multi-class classification one. The limitation of this approach is that it leads us to deal with an increased amount of classes.

In general, multi-label classifiers are learnt by assuming the existence of training sets in which each instance is associated with a precise set of labels. However, in practice, gathering such high quality information is not always feasible at a reasonable cost. In many problems, there is no ground truth for assigning unambiguously a label set to each instance, and the opinions of one or several expert have to be elicited. Typically, an expert will sometimes express lack of confidence for assigning exactly one label set. If several experts are consulted, some conflict will inevitably arise, which again will introduce some uncertainty in the labeling process. Thereby, in many real-world applications, we are facing situations where we have to deal with imperfect labeled instances and to handle imprecisions and uncertainties in data labeling.

Three original methods for multi-label learning will be exposed in this thesis. All are based on the k-nearest neighbor rule widely used in Machine Learning due to its simplicity and effectiveness at the same time, but associated with a different theoretical framework among probability, possibility, and evidence theories.

The first method relies on the binary relevance approach, while overcoming its label independence assumption. This methods addresses the problem of representing correlation between classes in a probabilistic framework. The classification of new instances is carried out by exploiting statistical information extracted from the nearest neighbors of the instances to classify and through Bayesian inference.

The two other methods address mainly the problem of learning from data with imprecise labels, and they have the ability to handle multi-labeled data directly. The basic idea of these two methods is to consider the class labels of multi-labeled instances as set-valued variables, i.e. variables that can assume multiple values simultaneously, and to use formalisms for manipulating imprecise and uncertain information about such variables. Possibility and Dempster-Shafer evidence theories provide formalisms devoted to to handle incomplete knowledge.

A possibilistic formalism for the expression of statements involving veristic variables has been proposed in [119]. Veristic variables can be viewed as fuzzy set-valued variables. This formalism provide different types of veristic statements and different distributions allowing us to encode any piece of knowledge about veristic variables. As alternative to this approach, we study the problem of handling partial knowledge on set-valued

variables using the evidence theory. The classical approach consists in considering a set-valued variable taking values in a universe Ω as a single-valued variable on the power set 2^Ω of Ω. If we want to express imprecise information about such a variable, we will have to manipulate subsets of 2^Ω. As there are $2^{2^{|\Omega|}}$ of these subsets, this approach rapidly becomes intractable as the cardinality $|\Omega|$ of Ω increases, due to the double-exponential complexity involved.

A main contribution of this thesis is the definition of an evidence formalism for representing uncertainty on set-valued variables using the Dempster-Shafer theory of belief functions [93]. In this formalism, instead of considering the whole power set of 2^Ω to express imprecise information about set-valued variables defined on Ω, only a class $\mathcal{C}(\Omega)$ of subsets of 2^Ω will be considered which, endowed with set inclusion, has a lattice structure. Most concepts of Dempster-Shafer theory can be generalized in this setting. This formalism will be shown to be rich enough to express evidence about set-valued variables with only a moderate increase of complexity as compared to the classical case of single-valued ones.

We will show applications of the veristic variable theory and the proposed evidence formalism for set-valued variables to multi-label learning, conjunctively with the k-nearest neighbor principle.

Organisation

This thesis is structured in five main chapters. Chapter 1 will summarize the state of the art on the multi-label learning problem, and report some related learning problems. The probabilistic method for multi-label classification will be presented in Chapter 2. A general overview on Bayesian learning will be first introduced, and the binary relevance approach as well the crucial issue of label correlation will be then discussed. Chapter 3 will present the possibilistic multi-label classifier. This chapter begins by a review of possibility and fuzzy set theory. The veristic variable theory will be then introduced. At the end of the chapter, the veristic k-nearest neighbor rule will be presented. The evidence-based multi-label learning will be detailed in Chapter 4. After introducing the basics of belief function theory, the proposed evidence formalism for representing and handling uncertainty on set-valued variables will be explained. Chapter 4 ends with an application of the proposed set-valued evidence formalism on multi-label learning.

Chapter 5 will describe the experimental results of the proposed multi-label classifiers on several real-world datasets, in the case of precise and imprecise data. Comparisons with some state-of-the-art methods, over different evaluation metrics, will be reviewed. General conclusion and perspectives will conclude the report.

1

Multi-label learning

Summary

Several methods have been proposed in the literature to deal with the task of multi-label learning, which is required by many modern applications such as semantic scene classification and video annotation. The common point between these methods is that they transform a multi-label learning problem into one or more single-label learning problems. The transformation is based on three approaches: Binary Relevance, Label Ranking and Label Powerset.

The Binary Relevance approach consists in training a binary classifier for each possible class in order to separate the instances belonging to that class from the others. The output of the multi-label classifier is the union of the decisions given by the binary ones.

The Label Ranking approach consists first in ranking all possible classes in decreasing order of relevance to an instance to classify, and then splitting the ordered set of classes into subsets of relevant and non relevant classes for that instance.

The Label Powerset approach consists in training a multi-class classifier for which, each combination of labels that exists in the given training set is considered as a new class. The most probable class is predicted for each new instance, which represents now a set of labels.

These different multi-label learning approaches will be discussed in this chapter, highlighting their positive and negative aspects.

Résumé

Plusieurs méthodes ont été proposées dans la littérature pour traiter la problématique d'apprentissage multi-label devenant de plus en plus requise par de nombreuses applications modernes telles que la classification d'images selon la sémantique, et l'annotation de vidéos. Le point commun entre ces méthodes est qu'elles consistent à transformer le problème d'apprentissage multi-label en un ou plusieurs problèmes d'apprentissage mono-label. La transformation est basée sur trois approches : Binaire, Classement de Labels, et Combinaisons de Labels.

L'approche Binaire constitue un classifieur binaire pour chaque classe possible afin de séparer les individus appartenant à cette classe des autres individus. La sortie du classifieur multi-label est déterminée par combinaison des sorties des différents classifieurs binaires.

L'approche de Classement de Labels consiste d'abord à classer les différentes classes par ordre décroissant de pertinence pour un individu à classifier, et de diviser ensuite l'ensemble ordonné de classes en un sous-ensemble de classes pertinentes, et en un autre sous-ensemble de classes non pertinentes.

L'approche de Combinaisons de Labels consiste à entraîner un classifieur multi-classes, tel que chaque combinaison de labels qui existe dans l'ensemble de données d'apprentissage est considérée comme une nouvelle classe pour ce classifieur. La classe la plus probable est attribuée à chaque individu à classifier, cette classe représente désormais un ensemble de labels.

Ces différentes approches d'apprentissage multi-label seront abordées dans ce chapitre, en soulignant leurs aspects positifs et négatifs.

1.1 Introduction

Machine learning is the field of research that concentrates on the formal study of learning systems. Over the years, machine learning has grown rapidly to become a highly interdisciplinary field overlapping with more traditional disciplines such as computer science, statistics, artificial intelligence, optimisation theory and many other disciplines of science and mathematics [42][77][6]. The majority of the work in Machine Learning concerns three principal learning frameworks: supervised, unsupervised, and semi-supervised.

In the supervised framework [62], the learning is performed on labeled examples (also called instances or samples)in order to define a function that predicts correctly the labels of the training examples; the performances of the obtained function are evaluated according to its ability of generalization when predicting the labels of examples not in the training set. If the labeling of the training examples is categorical (discrete labels or classes), the learning task is called *classification*. If the labeling is numerical, the task is called *regression*.

In unsupervised learning [54] the examples are not labeled, i.e., there are no supervised target outputs. The algorithm attempts to learn the structure of the given data and to organize them. The typical unsupervised learning problem is *clustering* that identifies groups of examples that have characteristics in common and are cohesive and separated from each other.

In semi-supervised learning problem [9], the training data is a mixture of both labeled and unlabeled examples. In fact, the acquisition of labeled data for a learning problem is not always feasible; the cost of the labeling process may be relatively high and it requires the efforts of several experts. In contrast, unlabeled data may be relatively easy to acquire.

1.1.1 Multi-label classification

In this work, we are interested in the classification task for *multi-label learning* [74][91][11]. Given a set of n training examples $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subseteq \mathbb{X}$, where $\mathbb{X}$ denotes the domain of instances, and a set of target classes $\mathcal{Y}$, traditional single-label classification assign each training instance $\mathbf{x}_i$ to a single label $y_i \in \mathcal{Y}$, and the goal is to learn a single-label classifier $\mathrm{h} : \mathbb{X} \longrightarrow \mathcal{Y}$ that predicts the class label of unseen examples. If there are only two

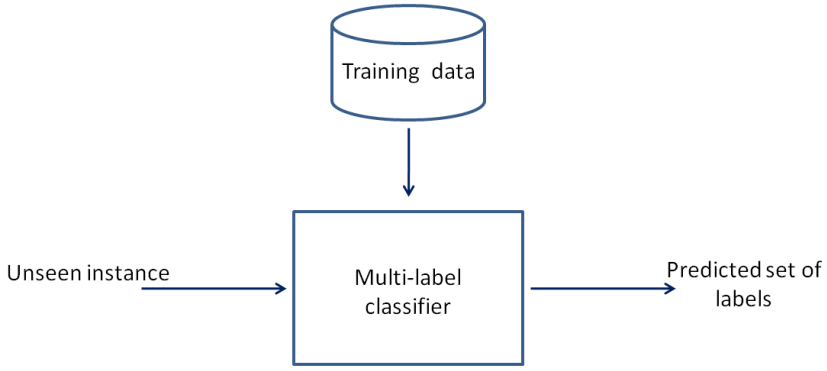

Figure 1.1: Multi-label learning system.

possible classes, the learning problem is called *binary* classification problem. When the number of classes is greater than two, it is called *multi-class* classification problem. In the case of multi-label classification problems, each training instance is assigned to a set of classes $Y_i \subseteq \mathcal{Y}$, and the goal of multi-label learning is to learn a multi-label classifier $\mathcal{H} : \mathbb{X} \longrightarrow 2^{\mathcal{Y}}$ that predicts a set of labels for each instance to classify (see Figure 1.1). Note that for the traditional single-label classification task, the target classes are disjoint and exclusive and each example belong to one and only one class, while for the multi-label classification task, the target classes are not exclusive and an example may belong to an unrestricted set of classes instead of exactly one class. Figure 1.2 shows an example of a classification problem with two classes that overlap in the feature space. In the case of single-label learning, the overlapping classes cause classification errors, while in multi-label learning, the classes overlap by *definition* in the selected feature space. For multi-labeled data, the membership of an example to more than one class is not due to ambiguity (*fuzzy* membership), but to multiplicity (*full* membership) [8]. Note that the traditional supervised learning (binary or multi-class) can be regarded as the special cases of multi-label learning, where the labels associated with each instance are restricted to be unique.

In multi-label learning problems, classes are usually correlated and a key challenge for a multi-label classification method is its ability to exploit correlation information among different classes. For example, in text categorization, a document is unlikely to be labeled as *politics* if we know that it belongs to class *entertainment*. In scene

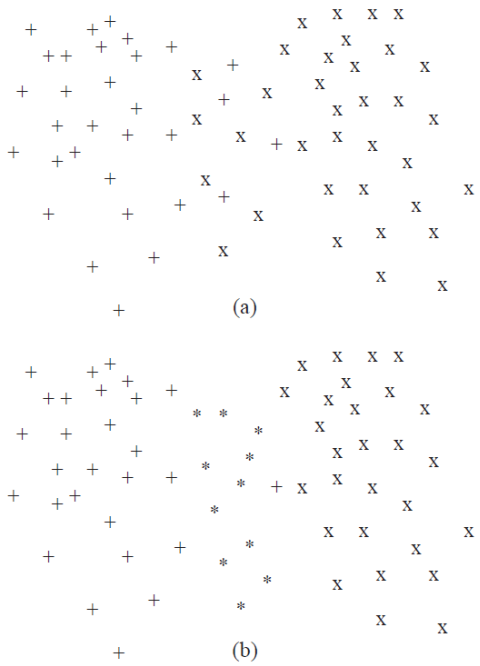

Figure 1.2: Single-label classification problem with two overlapping classes (a), and multi-label classification problem with data (∗) belonging simultaneously to the two possible classes (b).

classification, the probability that an image belongs to class *sunset* is high if this image is annotated with label *sea*. Thus, taking label correlations into consideration is a crucial requirement for the good performance of any multi-label classification method.

1.1.2 Applications

Multi-label learning methods for classification are required by many modern applications where it is quite natural that instances belong simultaneously to several classes. Hereafter, we will describe some of these applications.

With the rapid growth of online information and the ubiquity of textual data, *text categorization* has become an important task for many applications that require techniques for handling and organizing text data [75][92]. Document filtering, browsing and searching on the web and in large collections of documents, and email classification are such applications [68]. Due to the multi-topic nature of documents, multi-label learning methods seem to be adapted for text categorization [91][16][44][45]. For example,

Figure 1.3: Text categorization

Figure 1.3 shows an electronic document that deals with the olympic games and their influence and consequence on the hosting country. This document can be considered as belonging to the following predefined topics: *sport*, *politics*, *society* and *economy*.

Scene classification is a fundamental problem in image processing and a major task in computer vision [84][67]. It has received considerable attention in the recent past years, especially with the development of digital cameras. Scene classification is required for organisation of image collections; it has been explored in content-based image retrieval, and used to improve the performance of object recognition systems [106] [10]. Multi-label learning is required in semantic scene classification where a natural scene may contain multiple objects [8][111]. Figure 1.4 shows an example of an image labeled by three semantic classes: *mountain*, *trees* and *lake*.

Other multimedia applications for multi-label learning are *music classification* and *video annotation*. With the expansion of digital music libraries, the need for classification, retrieval and content-based searching tools through these files is becoming more and more apparent [69]. For example, music listeners may be interested in browsing their music by mood [58][71]. Due to the fact that a song can evoke more than one emotion at the same time, such as *amazed*, *happy* and *excited*, multi-label classification of music according to emotions has been investigated in recent years [101][114]. In ad-

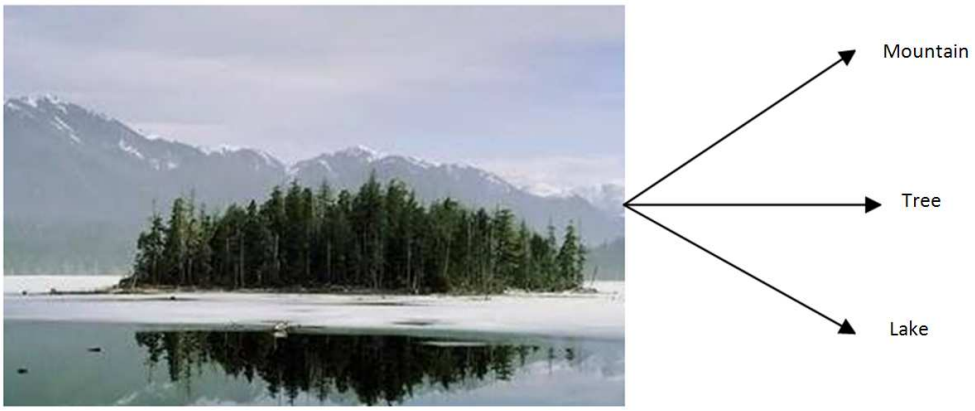

Figure 1.4: Semantic scene classification

dition, video annotation or tagging task is required for browsing and retrieval queries with the large increase of video data [79]. The annotation task is a multi-label problem where a film can be annotated with several labels or tags, such as *drama*, *fantasy* and *romance*, as shown in Figure 1.5 [87][25].

In addition to the above applications, multi-label learning has also proved to be useful in *bioinformatics* and especially for protein function prediction, where each protein may be associated with multiple functional labels such as *metabolism*, *energy* and *cellular biogenesis* [2][13].

1.2 Different approaches to multi-label learning

Let $\mathbb{X}$ denote the domain of instances, and $\mathcal{Y} = \{\omega_1, \ldots, \omega_Q\}$ the finite set of labels. Let $\mathcal{D} = \{(\mathbf{x}_1, Y_1), \ldots, (\mathbf{x}_n, Y_n)\}$ be a dataset composed of n multi-labeled object $(\mathbf{x}_i, Y_i)$, where $\mathbf{x}_i \in \mathbb{X}$ and $Y_i \subseteq \mathcal{Y}$. This dataset will be used to build a multi-label classifier $\mathcal{H}$ that defines a mapping from the domain of instances $\mathbb{X}$ to the power set $2^{\mathcal{Y}}$ of $\mathcal{Y}$.

Several methods have been proposed in the literature for multi-label learning. In general, these methods consists in transforming the Multi-label Classification problem (MLC) into one or more Single-Label Classification problems (SLC) [102]. The state-of-the-art methods are usually based on three approaches: *Binary Relevance*, *Label Powerset*, and *Label Ranking*.

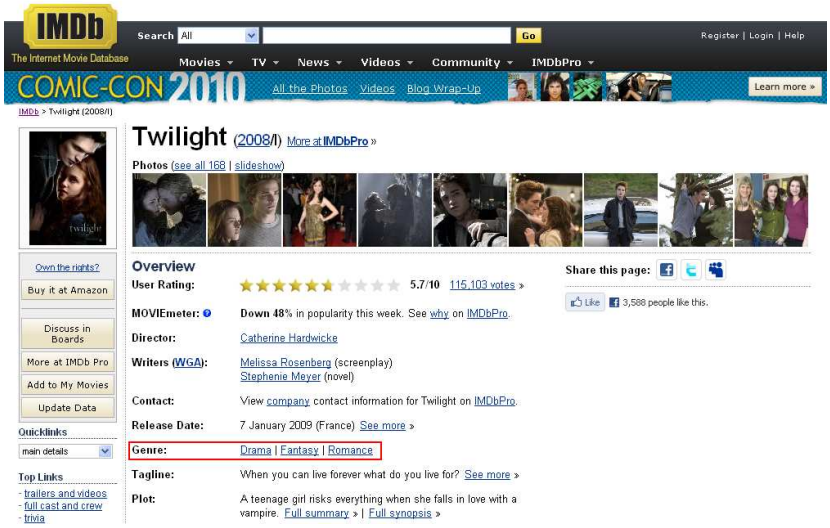

Figure 1.5: Film annotation

Hereafter, we will explain the different multi-label learning approaches, highlighting their positive and negative aspects.

1.2.1 Binary Relevance

The Binary Relevance (BR) method is the simplest and most commonly used approach to multi-label classification [59][121][83]. The BR approach transforms the multi-label learning problem with Q possible classes into Q single-label classification problems: $\mathrm{SLC}_1, \ldots, \mathrm{SLC}_\mathrm{Q}$. Each single-label classification problem SLC_q consists in separating the instances belonging to class ω_q from the others. This problem is solved by training a binary classifier h_q (0/1 decision) where each instance $\mathbf{x}_i$ in the training dataset $\mathcal{D}$ is considered as *positive* if it belongs to the class ω_q ($Y_i \ni \omega_q$), and *negative* otherwise. Given an instance $\mathbf{x}$ to classify, the output of the multi-label classifier $\mathcal{H}$ is the union of the decisions given by the binary classifiers $\mathrm{h}_1, \ldots, \mathrm{h}_\mathrm{Q}$:

$$\mathcal{H}(\mathbf{x}) = \{\omega_q \in \mathcal{Y} | \mathrm{h}_q(x) = 1\}.$$

The BR approach is intuitive, simple and it has low computational complexity. Given a constant number of training examples, the complexity of BR approach scales linearly with the number of possible labels. However, the BR method does not take

into account correlations between labels. Each binary problem SLC_q $(q = 1, \ldots, \text{Q})$ is independent from the other problems, and is solved separately by running the single-label classifier h_q, in serial or parallel to the other binary classifiers, on the training dataset $\mathcal{D}$. Due to the implicit assumption of label independence, the BR-based methods may be penalized and their performances may be poor, especially when applied to multi-label learning problems in which the labels are highly correlated.

Any known single-label classifier can be used for the binary classification subproblem. A set of binary support vector machine (SVM) [107] classifiers were used for multi-label learning in text categorization [57] and semantic scene classification [8]. In [70], active learning for multi-label classification using an ensemble of binary SVMs has been presented. Active learning is a mechanism that aims at minimizing the number of labelled training data while maintaining a good classification performance [15]. In practice, active learning is very useful in situations where data are expensive or difficult to collect. In [47], an improvement of the BR-based approach using SVM as binary classifier has also been proposed. The improvement is obtained by tuning the margins of the SVMs to account for classes that overlap. In the first iteration, the ensemble of Q SVMs classifiers is trained. For each trained SVM, the misclassified training instances that are close or within a threshold distance from the learnt hyperplane are removed. Then, the ensemble of the SVMs classifiers is re-trained. By removing the points that are very close to the resultant hyperplane for a SVM classifier, the authors show that one can train a better hyperplane with a wider margin and thus improve the classification accuracy. Another way proposed in [47] to improve the margin is to completely remove the training instances belonging to *confusing* classes. Confusing classes are detected using a confusion matrix learnt using any moderately accurate yet fast classifier on a held out validation dataset. If the percentage of instances of class ω_q that were misclassified as belonging to class ω_r is above a threshold, we prune away the instances of class ω_r when training the binary SVM classifier corresponding to ω_q.

Using the k-nearest neighbor (k-NN) algorithm, a multi-label classification method named MLkNN has been proposed in [129]. Each binary classifier h_q is implemented by means of a combination of k-NN and Bayesian inference. Given an instance $\mathbf{x}$ to classify, its k nearest neighbors in the training dataset are identified ; those belonging to class ω_q are considered as positive for h_q, and the rest as negatives. The classification of $\mathbf{x}$ by the binary classifier h_q is determined by computing the posterior probability

of "$\mathbf{x}$ belong to ω_q" based on the prior probability of ω_q and statistical information gained from the label sets of the neighboring instances. This method will be presented in greater detail in the next chapter.

Few methods have been proposed in the literature to remedy the disadvantage of the BR approach of ignoring label correlations. In [47], multi-label learning is achieved by creating a two-stage classification process and by using labels in the feature space. In the training stage, a set of Q binary classifiers $\mathrm{h}_1, \ldots, \mathrm{h}_\mathrm{Q}$ are run on the training dataset $\mathcal{D}$ in the first classification process. The predictions of each binary classifier are used to extend the original dataset with Q additional *label features*. Each object $(\mathbf{x}_i, Y_i)$ in $\mathcal{D}$ is transformed into a meta-object $(\mathbf{x}'_i, Y_i)$, where $\mathbf{x}'_i = (\mathbf{x}_i, \mathrm{h}_1(x_i), \ldots, \mathrm{h}_\mathrm{Q}(x_i))$. The second classification process consists in the training of new Q binary classifiers using the meta-objects. Given a new instance to classify, the binary classifiers of the first classification process are used and their outputs are appended to the initial features to form a meta-instance. This meta-instance is then classified using the binary classifiers of the second process. The correlations between labels is taken into account by this approach through the label feature stacking.

Following a similar idea, a *classifier chain* (CC) model involving Q binary classifiers has been introduced in [89]. The classifiers are linked along a chain. At each link, the feature space of the training data is extended with the 0/1 label associations of all previous links. More precisely, given an object $(\mathbf{x}_i, Y_i)$ in $\mathcal{D}$, the labeling of $\mathbf{x}_i$ can be represented by the category vector $\mathbf{y}_i \in \{0,1\}^\mathrm{Q}$, where its q-th component $\mathbf{y}_i(q)$ takes the value 1 if $\omega_q \in Y_i$ and 0 otherwise. At link q, the training data is transformed into single-label data in the following way: each element $(\mathbf{x}_i, Y_i)$ is transformed into $((\mathbf{x}_i, \mathbf{y}_i(1), \ldots, \mathbf{y}_i(q-1)), \mathbf{y}_i(q))$, and the binary classifier h_q is trained on the transformed data. Given an instance $\mathbf{x}$ to classify, the classification process is performed by moving along the chain from the first link to the last one. Example $\mathbf{x}$ is first classified by h_1, and the 0/1 prediction about the membership or not to class ω_1 is appended to its feature vector in order to be classified by h_2, and so on. The dependencies between labels are taken into account by passing label information between the chain classifiers. However, it is clear that the classification performance depends on the order of the chain. Therefore, an ensemble of chain models (ECC) is used to create different random chain orderings.

In [55], a second process is added upon the BR approach to derive a low-dimensional subspace share among multiple labels. The idea behind this approach is that, when two labels are correlated, the corresponding instances shared some characteristics in the feature space. For example, when predicting the topics of documents, there is a relation between authorship and topics, since a given author may usually write on known topics. In the proposed framework, a binary classifier is constructed for each class in order to discriminate it from the other classes. The input data are projected onto a low-dimensional subspace using a common transformation for all classes, and this low-dimensional projection is combined with the original representation to produce the final prediction.

1.2.2 Label Ranking

A second approach consists in transforming the multi-label learning task into a label ranking problem. A label ranking (LR) method predicts a ranking of all possible labels in decreasing order of relevance to a query instance. Afterwards, a post-processing is required in order to determine the output of the multi-label classifier. In the multi-label case, the *topmost labels*, and not only the top label, are related to the instance to classify. Thus, the goal of the post-processing is to provide a *zero-point* that splits the ordered set of labels into subsets of relevant and non-relevant labels for the query instance. The LR approach does not explicitly model the correlations among labels. Another problem is that it is difficult to determine into how many labels a particular instance should be classified. The prediction or the ranked set splitting is usually done by a thresholding technique.

A straightforward LR-based approach learns a multi-label classifier $\mathcal{H} : \mathbb{X} \longrightarrow 2^{\mathcal{Y}}$ via a scoring function $f : \mathbb{X} \times \mathcal{Y} \longrightarrow \mathbb{R}$ that assigns a real value (score) to each instance/label couple $(\mathbf{x}, \omega) \in \mathbb{X} \times \mathcal{Y}$. The score corresponds to the probability that the class ω is relevant to the instance $\mathbf{x}$. In addition, given any instance $\mathbf{x}$ with its known set of labels $Y \subseteq \mathcal{Y}$, the scoring function f is supposed to give larger scores for labels in Y than it does for those not in Y. In other words, $f(\mathbf{x}, \omega_q) > f(\mathbf{x}, \omega_r)$ for any $\omega_q \in Y$ and $\omega_r \notin Y$, for each object $(\mathbf{x}, Y)$. The scoring function f allows us to rank the different labels according to their scores. For an instance $\mathbf{x}$, the higher the rank of a label ω, the larger the value of the corresponding score $f(\mathbf{x}, \omega)$. The output of the multi-label

classifier $\mathcal{H}$ is determined by selecting the labels from the top of the ranking using some threshold value $t \in \mathbb{R}$:

$$\mathcal{H}(\mathbf{x}) = \{\omega \in \mathcal{Y} | f(\mathbf{x}, \omega) \geq t\}.$$

The threshold value can be determined by cross-validation or heuristically [40]. For example, in [100], the threshold value is fixed by minimizing the difference of *label cardinality* between training and test datasets. The label cardinality of a given dataset is defined as the average number of labels per instance.

Another ranking-based approach learns a multi-label classifier not via a scoring function, but via pairwise comparisons. A recent method is the ranking by pairwise comparison (RPC) introduced in [53]. The multi-label learning problem is transformed into a number of binary problems. An independent binary classifier h_{qr} is trained for each pair of labels $(\omega_q, \omega_r) \in \mathcal{Y}^2$, $1 \leq q < r \leq \mathrm{Q}$, in order to separate the instances with label ω_q from those having label ω_r. Thus, a total number of $\mathrm{Q}(\mathrm{Q}-1)/2$ is required. In the classification phase, a new instance is submitted to each binary classifier h_{qr}, and the prediction is interpreted as a vote for either ω_q or ω_r. The labels with the highest number of votes are proposed as a final prediction for the query instance via thresholding. Instead of learning a predictor for the correct threshold, a modified multi-label ranking-based approach, called calibrated label ranking (CLR), has been presented in [43]. In this method, the zero-point at which the learned ranking is split into sets of relevant (or positive) and irrelevant (or negative) labels is determined automatically. In fact, CLR incorporates an additional *virtual label* ω_0 in the ranking process, which calibrates the ranking by splitting it into a positive and a negative part. In addition to the pairwise classifiers h_{qr}, $1 \leq q < r \leq \mathrm{Q}$ as in RPC approach, CLR adds Q classifiers h_{q0}, $1 \leq q \leq \mathrm{Q}$ that separate each class ω_q from the virtual class ω_0. Each classifier h_{q0} is learned by considering all instances belonging to ω_q as positive, and the remaining instances, considered as belonging to the virtual class, as negatives. Thus, the binary classifiers h_{q0} are trained as in the BR approach. In CLR, we have to train $\mathrm{Q}(\mathrm{Q}+1)/2$ pairwise classifiers. For the classification of a new instance, all labels ranked above the virtual label ω_0, i.e., receiving more votes than ω_0, are assigned to the instance.

Hereafter, we will present some state-of-the-art methods based on the ranking approach.

In [91], a boosting-based system for multi-label learning and especially for text categorization, named BoosTexter, has been introduced. The system is based on two

extensions of the ensemble learning method AdaBoost [41], where a set of weights over all instance/label pairs is maintained. As boosting progresses, training instances and their corresponding labels that are hard to predict correctly get incrementally higher weights, while instances and labels that are easy to classify get lower weights. The goal of the first extended learning algorithm is to predict a set of correct ones for a query instance. In the second extension, the goal is to design a classifier that ranks all labels so that the correct labels for training instances will receive the highest ranks.

In [39], multi-label ranking approach based on support vector machines (SVM) has been presented. The authors define a cost function and a special multi-label margin and then propose an algorithm named Rank-SVM based on a ranking system combined with a label set size predictor. The set size predictor is computed from a threshold value that differentiates the relevant labels from the others. The value is chosen by solving a learning problem. The goal is to minimize the Ranking Loss, defined as the average number per instance of label pairs that are not correctly ordered, while having a large margin. Rank-SVM uses kernels rather than linear dot products, and the optimisation problem is solved via its dual transformation.

Following a similar line of reasoning, a multi-class multi-label perceptrons algorithm has been presented in [18] where one perceptron is trained for each possible label. The classifiers are not trained independently, but in such a way that they collectively produce a reasonable ranking for a given ranking loss function. In [76], pairwise multi-label perceptrons have been introduced. Based on the RPC approach, one perceptron is trained for each pair of labels independently of other perceptrons. A calibrated version of the pairwise multi-label perceptrons, based on the CLR approach, has been presented in [43].

In [128], a neural network algorithm for multi-label learning, named BP-MLL, is presented. BP-MLL is a single-hidden feed-forward neural network with Q output neurons, each one corresponding to one of the possible classes. The parameters of the proposed neural network are learnt by minimizing a specific error function different from the simple sum-of-squares function used in the classical single-label case. The error function, defined as the difference between the actual and the desired outputs of the neural network, is adapted for the purpose of multi-label learning in such a way that, given an instance $\mathbf{x}$, the labels assigned to $\mathbf{x}$ should be ranked higher than those not assigned to this instance.

In [126], an adaptation of the traditional radial basis function (RBF) neural network for multi-label learning is presented. It consists of two layers of neurons: a first layer of hidden neurons representing basis functions associated with prototype vectors, and a second layer of output neurons related to all possible classes. The proposed method, named ML-RBF, first performs a clustering of the instances corresponding to each possible class; the prototype vectors of the first-layer basis functions are then set to the centroids of the clustered groups. In a second step, the weights of the second-layer are fixed by minimizing a sum-of-squares error function. The output neuron of each class is connected with all input neurons corresponding to the prototype vectors of the different possible classes. Therefore, information encoded in prototype vectors of all classes is fully exploited when optimizing the connection weights and predicting the label sets of unseen instances.

Remark that some BR-based multi-label classifiers can also be considered as LR-based ones, because they are able to provide scoring functions for ranking.

1.2.3 Label Powerset

Given the set of possible labels $\mathcal{Y}$ and a set $\mathcal{D}$ of n training data for a given multi-label learning problem, the Label Powerset (LP) approach considers each subset of $\mathcal{Y}$ that exists in the training dataset $\mathcal{D}$ as a different class for a single-label classifier. The multi-label classification problem is then transformed into a multi-class classification problem, with a number of classes at most equal to $\min(2^{\mathrm{Q}}, n)$. The LP method has the advantage of taking label correlations into consideration. There are no binary classifiers to be learnt independently for each label. Another advantage of this approach is that there is no threshold to be tuned, and the LP-based methods output directly a set of labels. In fact, for each unseen instance, the most probable class that represents now a set of labels is predicted. Nevertheless, one of the drawbacks of the LP approach is that it may lead to *imbalanced* datasets with a large number of classes and few examples per class, which makes the learning process difficult and poses computational complexity problems with the increasing number of labels. Another disadvantage is that LP can only predict label sets available in the training set when classifying new instances.

Few approaches have been presented in the literature in order to deal with the aforementioned negative aspects of LP principle, while preserving its advantage of taking label correlations into consideration. In [104], a method named RAKEL works by

randomly breaking the set of labels $\mathcal{Y}$, supposed to be large, into a number of label sets $Z_i \subseteq \mathcal{Y}$ having a small size as compared to the size of $\mathcal{Y}$, and training a LP-based multi-label classifier for each of the label sets. More precisely, for each small-sized label set Z_i, a single-label classifier h_i, having as class values all the subsets of Z_i that are found in the training set, is trained. The training set for h_i, denoted as $\mathcal{D}_i$ is deduced from the original dataset $\mathcal{D}$ by replacing the label sets of training instances by their intersections with $Z_i : \mathcal{D}_i = \{(\mathbf{x}_j, Y_j \cap Z_i),\ j = 1, \ldots, n\}$. In particular, this may lead to instances labeled by the empty set. These instances are not excluded when training h_i, and one has to consider the empty label set as another class value for the single-label classification task of h_i. For the classification of a new instance, the decisions of the LP-based classifiers are gathered and combined, usually by a voting process. This method can predict a label set that was not present in the training set, because the final output of the multi-label classifier is computed from the predictions of the different single-label classifiers. The number of random label sets to be considered and the size of these sets have to be tuned heuristically, which is computationally expensive.

In [88], a new extension of the LP approach has been proposed in order to reduce its complexity. The new approach, called Pruned Sets (PS), works as LP but only the label combinations, subsets of $\mathcal{Y}$, which *frequently* occur in the training dataset $\mathcal{D}$ are considered as class values for the single-label classifier. The pruning operation is controlled by a parameter that indicates how often a label combination must occur in the training set in order not to be pruned and then to be considered as a new class value. For minimal information loss, a post-pruning step is added in order to break up the pruned label combinations into more frequently occurring label sets, and then reintroduce the pruned instances in the training process of the single-label classifier.

Based on the LP-approach, a probabilistic generative model for multi-label text categorization has been introduced in [74]. According to this method, each label (topic) generates different words, and a document is produced by a mixture of the word distributions of its labels. Given a set of classes, each document is generated by a mixture of word distributions and mixture weights, where the weight of classes not belonging to this set are forced to be zero. The parameters of the model are determined using a maximum a posteriori estimation from a collection of labeled training data. The Expectation-Maximization (EM) is used to determine the parameters that cannot be estimated directly from the training dataset. For the classification of a new document,

the Bayes rule is employed in order to calculate the posterior probability of each set of classes and select the most likely set given the document. A similar word-based mixture model is presented in [105], where two parametric mixture models are proposed. Finally, a maximum entropy model is introduced in [131] in order to capture the pairwise class correlation by adding second order constraints.

1.3 Related learning problems

The different approaches for multi-label learning that have been presented in the previous section concern the so-called *flat* multi-label classification task. However, in some problems, the classes are hierarchically organized, imposing the constraint that when an instance is assigned to a certain class, it should also be assigned to all its superclasses [90]. This learning task is called *hierarchical multi-label classification*. The hierarchy of classes can be such that each class has at most one parent or superclass (tree structure) or such that classes may have multiple parents (direct acyclic graph structure) [108]. Examples of this kind of problems are found in several domains, including text classification [90] and functional genomics[4][7][13]. In [103], a method called HOMER transforms the multi-label classification problem with a large set of labels into a tree-shaped hierarchy of multi-label classification subproblems with a small number of labels. The hierarchical splitting of the set of labels is done using a modified k-means algorithm.

A distinction should be made between multi-label and *multiple-label* learning problems. Multiple-label learning [56] is a semi-supervised learning problem for single-label classification where each instance is associated with a set of labels but only one of the candidate labels is the true label for the given instance. For example, this situation occurs when the training data is labeled by several experts, and due to conflicts and disagreements between the experts, a set of labels, instead of exactly one label, will be assigned to some instances. The set of labels of an instance contains the decision (the assigned label) made by each expert about this instance. It means that there is an ambiguity in the class labels of the training instances.

Multi-instance learning for classification is an another variation of supervised learning problems where each training object is represented by a *bag* of instances (feature vectors) and is assigned a single label [24][72]. For example, this learning problem is

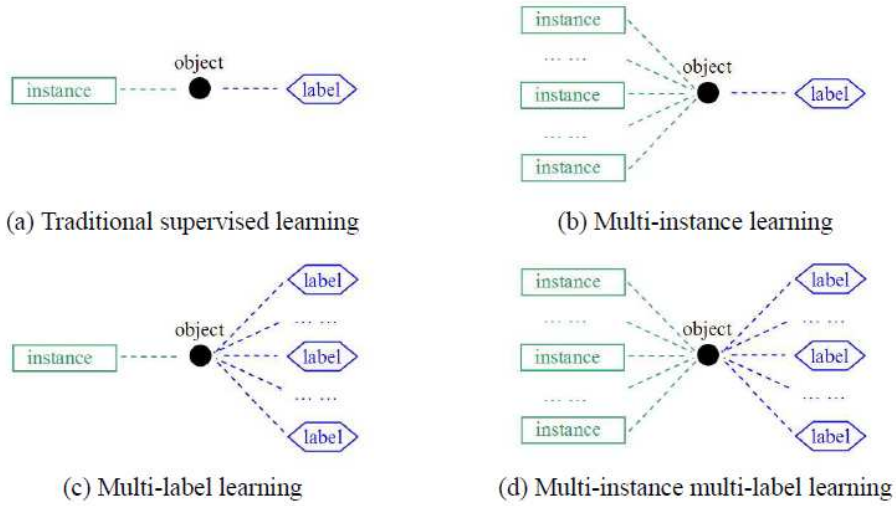

Figure 1.6: Different learning problems

encountered in drug activity prediction [24]. A molecule, qualified to make some drug, is originally small and works by biding to a larger protein molecule. It is known that the biding strength depends on the shape of the molecule. Each molecule has multiple possible shapes and it is generally unknown which of these shapes (one or more) cause the biding [130]. Thus, for this problem, we have to represent an objet by a bag of instances, each one describing a shape, and the bag is labeled as corresponding to a drug molecule or a non-drug one.

Another learning problem is *multi-instance multi-label* learning where each object is described by a bag of instances and is assigned a set of labels [127][94]. This learning problem combines the multi-instance and the multi-label learning tasks. Different real-world applications can be handled under this framework. For example, in text categorization, each document can be represented by a bag of instances, each instance representing a section of this document, while the document may deal with several topics at the same time, such as *culture* and *society*.

Figure 1.6 illustrates the different learning frameworks: traditional supervised learning for single-label classification, multi-instance learning, multi-label learning and multi-instance multi-label learning.

In this work, we focused on the study of *non-hierarchical* multi-label learning prob-

lems, where each training object is described by *a single* feature vector and may belong to *several* classes at the same time.

1.4 Contributions

Three original methods for multi-label learning will be exposed in this thesis, using the k-nearest neighbor rule as base classifier.

The first method, called DMLkNN for Dependent Multi-Label k-NN, is a *probabilistic* multi-label classification method able to exploit information about label interdependencies, which is very important for the success of multi-label classification techniques. This method generalizes the state-of-the-art MLkNN algorithm by relaxing the assumption of label independence. A maximum a posteriori (MAP) estimation is used in order to determine the proper set of labels to be assigned to a test instance $\mathbf{x}$, according to statistical information extracted from the labeling of the nearest neighbors. For each $\omega_q \in \mathcal{Y}$, the numbers of neighboring instances belonging to each possible class are used in order to compute the posterior probabilities that $\mathbf{x}$ belongs and does not belong to ω_q. Depending on which of these probabilities is greater, we decide to assign or not the class ω_q to test instance $\mathbf{x}$. The decision is made independently for each label, but correlation between labels is exploited when computing the two aforementioned probabilities. When computing the posterior probabilities for ω_q, the frequency of occurrence of a label ω_r in the label sets of the neighboring instances will affect the membership of $\mathbf{x}$ to class ω_q.

We also propose two multi-label classification methods that are able to handle multi-labeled data *directly*. As we have seen above, most existing multi-label learning algorithms transform the multi-label classification problem into one or more single-label learning tasks and adapt conventional classifiers for the multi-label purpose. BR, LR and LP are the three common transformation approaches. In contrast, for the proposed methods, the multi-labeled data are not transformed into single-labeled ones, and thus there is no information lost in data labeling. The two direct multi-label classifiers are *intrinsically* able to capture any relation between labels.

Another motivation behind the two developed methods is that, when learning a multi-label classifier, we generally assume the existence of a labeled training set in which each instance is associated with a perfect *well-known* set of labels. However, in

practice, gathering such high quality information is not always feasible at a reasonable cost. However, in many real-world applications, we are facing situations where we have to deal with imperfect labeled instances and to handle imprecisions and uncertainties in data labeling. Such situations occur for exemple, when the data are labeled subjectively by one or many experts. Possibility [124][33] and evidence [93][99] theories provide frameworks for reasoning under uncertainty and make it possible to handle easily such complex problems.

In [119], a possibilistic framework has been proposed for the expression of statements involving *veristic* variables, which can also be called fuzzy *set-valued* variables. Veristic variables are variables that can assume simultaneously multiple values with different degrees. Four types of veristic statements allow us to represent any piece of knowledge about veristic variables: open positive, open negative, exclusive positive, and exclusive negative statements. In multi-label learning, the class label of each instance can be considered as a veristic variable, since the instance can belong simultaneously to more than one class. The veristic theory will be used to build a multi-label classifier called VERkNN. The labeling of each training instance $\mathbf{x}_i$ is represented by two distributions: a *verity* distribution containing positive information about the labels that should be assigned to $\mathbf{x}_i$, and a *rebuff* distribution encoding negative information about the labels that should *not* be assigned to that instance. The verity and rebuff distributions corresponding to the neighboring training instances are discounted depending on the distance to the instance to classify, and are then combined in order to determine the set of classes to assign to the unseen instance.

In evidence theory, a *frame of discernment* Ω is defined as the set of all possible *exclusive* solutions of a given problem, where each variable can have one and only one solution in Ω. In a multi-label learning problem, the label set Y of each instance $\mathbf{x}$ is a set-valued variable taking values in the set of all classes $\mathcal{Y}$. A straightforward approach to study the problem of multi-label learning under evidence theory is, of course, to define the frame of discernment Ω as the set of all subsets of $\mathcal{Y}$. Each label set Y that represents a set-valued variable on $\mathcal{Y}$ is then considered as a single-valued variable on the frame of discernment $\Omega = 2^{\mathcal{Y}}$. However, this approach often implies working in a space of very high cardinality, as the size of the frame of discernment is $|\Omega| = 2^{\mathrm{Q}}$. If we want to express imprecise information about Y, we will have to manipulate subsets of

Ω. As there are 2^{2^Q} of these subsets, this approach rapidly becomes intractable as the number of possible classes Q increases.

A major contribution of this thesis is the definition of an approach able to handle and represent uncertainty about set-valued variables using the Dempster-Shafer theory of belief functions [93] and with only a moderate increase of complexity. Our approach represents an alternative to veristic theory for manipulating set-valued variables. The proposed approach will be based on a simple representation of a class $\mathcal{C}(\mathcal{Y})$ of subsets of $\Omega = 2^{\mathcal{Y}}$ which, endowed with set inclusion, has a lattice structure. Using recent results about belief functions on lattices [49], we will be able to generalize most concepts of Dempster-Shafer theory in this setting. This formalism will be shown to allow the expression of a wide range of knowledge about set-valued variables, with only a moderate increase of complexity (from 2^Q to 3^Q) as compared to the usual single-valued case.

Using the belief function framework for set-valued variables, we will present an evidence-theoretic k-NN rule for multi-label learning called EMLkNN. For this method, each neighbor of an instance $\mathbf{x}$ to classify is considered as an item of evidence supporting certain hypotheses regarding the class label of that instance. The degree of support is defined as a function of the distance between the two examples. Each item of evidence is represented by two disjoint subsets of $\mathcal{Y}$, a subset of classes that surely apply to the unseen instance $\mathbf{x}$, and a subset of classes that surely do not apply to $\mathbf{x}$. The evidence of the k nearest neighbors is then pooled by means of a combination rule in order to estimate the set of labels of the unseen instance.

1.5 Conclusion

In this chapter, an analysis of the state-of-the-art of the multi-label learning task has been exposed. We have shown that there are three main approaches for multi-label learning: BR, LR, and LP. The basic idea of these approaches consists in transforming a multi-label learning problem into one or more single-label learning ones.

Different real-world applications requiring multi-label learning have been described, and related learning problems have been also summarized.

Finally, we have discussed the aim of this thesis. Exploiting label correlation, and handling imprecision in data labeling are the main motivations.

In the next chapters, the three methods for multi-label learning will be detailed.

Chapter 2

Bayesian approach for multi-label learning

Summary

In this chapter, we propose a Bayesian k-nearest neighbor rule for multi-label learning. This method is able to take into consideration the correlation between labels. In fact, in multi-labeled data, the membership of an instance to a given class, may provide information on the membership of that instance to another class. For example, if an image is assigned to class "Desert", one can deduce that the image should not belong to class "Lake". Each query instance is classified on the basis of statistical information extracted from its nearest neighbors. More precisely, the probability of the assignment of an instance to a certain class is estimated, from the training dataset, based on the number of neighbors belonging to that class and also the number of neighbors belonging to each of the other classes. Since that the size of the training set is usually limited, the posterior probabilities are computed based on the *approximate* number of neighbors belonging to each class existing in the neighborhood.

Résumé

Dans ce chapitre, nous présentons une méthode Bayésienne pour l'apprentissage multi-label basée sur la règle de k-plus proches voisins. Cette méthode est capable de prendre en considération les correlations entre les différentes classes. En fait, en ce qui concerne les données multi-étiquetées, l'appartenance d'un individu à une classe donnée,

peut donner une certaine information sur l'appartenance de ce même individu à une autre classe. Par exemple, si on associe la classe "Désert" à une certaine image, on peut en déduire que cette image n'appartient pas à la classe "Lac". Dans la méthode proposée, la classification de chaque nouveau individu est basée sur des informations statistiques extraites de ses plus proches voisins. Plus précisément, la probabilité d'appartenance d'un individu donné à une certaine classe est estimée à partir de la base d'apprentissage, en fonction du nombre de voisins appartenant à cette même classe et aussi en fonction du nombre de voisins appartenant à chacune des autres classes. Vue que la taille de données d'apprentissage est souvent limitée, les probabilités à posteriori sont calculées en fonction du nombre *approximatif* de voisins appartenant à chacune des classes existant dans le voisinage.

2.1 Introduction

Binary Relevance (BR) is the most common approach for multi-label learning. A binary classifier is trained to separate one class from the others. The outputs of the different binary classifiers are combined in order to determine the final output of the multi-label classifier. The binary classifiers tacitly assume the non-dependency between labels. This assumption is questionable in many multi-label learning problems. In general, multi-labeled data exhibit relationships between labels, and binary classifiers fail to capture this effect. For example, {*entertainment*, *music*} is more likely than {*entertainment*, *politics*}, because documents that are under the label *music* are more likely to have also label *entertainment* in their label sets than label *politics*. Despite of this limitation, the BR approach is simple and intuitive and has the advantage of having low computational complexity.

A Bayesian algorithm for multi-label learning will be presented in this chapter. The proposed method is derived from the k-nearest neighbor rule and is able to capture dependencies between labels. The classification of an instance is carried out by exploiting statistical information extracted from its k nearest neighbors and through Bayesian inference. This method is called DMLkNN and generalizes the MLkNN algorithm presented in [129]. The proposed method relies on the binary relevance approach, in the sense that a decision is made separately for each class, while overcoming the label independence assumption.

This chapter is organized as follows. In Section 2.2, a general overview about Bayesian classification will be presented. In Section 2.3, the well-known k nearest neighbor rule will be described. Label correlation in multi-label learning will be discussed in Section 2.4. Section 2.5 will present the proposed multi-label classification algorithm based on a Bayesian interpretation of the k-NN rule. An illustration on a simulated dataset will be reported in Section 2.6. Finally, Section 2.7 will conclude this chapter.

2.2 Bayesian rule for classical classification problems

Different probabilistic model specifications can be designed in order to address the classification problem. Bayesian classification methods are in general based on the Bayes theorem [51][42]. This is a generative approach to classification and it offers a useful conceptual framework. It allows us to develop practical learning algorithms

providing, based on a training set, prior knowledge and observed information about instances to classify.

The generative probability model for a Bayesian classifier can be described as follows. Let $\mathbf{x}$ be an instance to classify, and let H_b^q denote the hypothesis that $\mathbf{x}$ belongs to class $\omega_q \in \mathcal{Y}$ if $b = 1$, and the hypothesis that $\mathbf{x}$ does not belong to ω_q if $b = 0$. $\Pr(\mathrm{H}_1^q|\mathrm{E})$ represents the *posterior* probability that $\mathbf{x}$ belongs to ω_q given the observed *evidence* E that represents knowledge about the instance to classify. Based on the maximum a posteriori (MAP) rule, a Bayesian classifier h assigns $\mathbf{x}$ to the class with the maximum posterior probability. For the computation of $\Pr(\mathrm{H}_1^q|\mathrm{E})$, for each $q \in \{1, \ldots, \mathrm{Q}\}$, the posterior probability is decomposed into a *prior* probability $\Pr(\mathrm{H}_1^q)$ and a *likelihood* $\Pr(\mathrm{E}|\mathrm{H}_1^q)$, using the Bayes theorem:

$$\Pr(\mathrm{H}_1^q|\mathrm{E}) = \frac{\Pr(\mathrm{E}|\mathrm{H}_1^q)\Pr(\mathrm{H}_1^q)}{\Pr(\mathrm{E})},$$

where $\Pr(\mathrm{H}_1^q)$ is the probability that an instance belongs to class ω_q, $\Pr(\mathrm{E}|\mathrm{H}_1^q)$ is the probability of observing E giving that the instance belongs to ω_q, and $\Pr(\mathrm{E})$ is the probability of observing E. For example, suppose that we have a document classification problem with two possible classes ω_1 for *scientific* and ω_2 for *literary*. A training dataset contains 60% of scientific documents and 40% of literary ones. Consider the observation that 70% of all scientific documents contain the word "hypothesis" and 5% contain the word "literary". Let E represents the evidence of observing the word "hypothesis" in a document. The probability $\Pr(\mathrm{E})$ of observing this word is $\Pr(\mathrm{E}) = \Pr(\mathrm{E}|\mathrm{H}_1^1)\Pr(\mathrm{H}_1^1) + \Pr(\mathrm{E}|\mathrm{H}_1^2)\Pr(\mathrm{H}_1^2) = 0.7 \times 0.6 + 0.05 \times 0.4 = 0.44$. The probability $\Pr(\mathrm{H}_1^1|\mathrm{E})$ of a document containing "hypothesis" and belonging to class ω_1 is $0.7 \times 0.6/0.44 = 0.95$, while the probability $\Pr(\mathrm{H}_1^2|\mathrm{E})$ of such a document belonging to the literary class is $0.4 \times 0.05/0.44 = 0.05$.

Note that $\Pr(\mathrm{E})$ may be considered as a normalization factor that can be ignored in practice when computing the posterior probability corresponding to each class ω_q, as it does not depend on the classes. Thus, the output of the Bayesian classifier h is determined in general as follows:

$$\mathrm{h}(\mathbf{x}) = \omega_r, \text{ such that:}$$

$$r = \arg\max_{q=1\ldots\mathrm{Q}} \Pr(\mathrm{E}|\mathrm{H}_1^q)\Pr(\mathrm{H}_1^q),$$

the prior probabilities and the likelihoods being estimated from training data.

2.3 Nearest Neighbor classification

The Nearest neighbor (NN) rule is one of the simplest and most popular methods for statistical learning [17]. This is an instance-based classifier that has been shown to be very effective in many classification problems [37][19]. The intuition is simple. Giving a single-labeled training set, the classification of a query instance $\mathbf{x}$ is performed by assigning it the label of the least distant training pattern according to some distance measure. The voting k-nearest neighbor rule, with $k \geq 1$, is a generalization of the NN approach where the most frequent class occurring in the k neighbors of $\mathbf{x}$ is predicted. The voting k-NN rule is less sensitive to noise on the available training data.

Clearly, the performances of the k-NN rule depend on the distance metric $\mathrm{d}(.,.)$ used to identify nearest neighbors, and the number k of neighbors to be considered for the classification of unseen instances [38]. Note that, usually, when feature variables are not of comparable units and scales and there is a great difference in the range of their different values, distance metrics implicitly assign greater weight to features with large ranges than those with small ones. In such cases, feature normalization is recommended to approximately equalize ranges of the features such that they will have the same effect on distance computation.

The Euclidean metric is the most popular distance function and it is widely used in k-NN classification. This metric, however, does not exploit any statistical properties and information that can be extracted and estimated from the training data. Many researches have been focused on the definition of distance metrics to improve the k-NN classification. Ideally, the distance metric should be *locally* adapted to the classification problem under study, and thus should be learnt a metric from the labeled training data. In [52], local linear discriminant analysis is used to estimate an effective metric for computing neighborhoods. The idea is to locally determine feature relevance for each query instance $\mathbf{x}$, so that its neighborhood gets an ellipsoidal shape elongated along the true decision boundary (the most relevant feature), and flattened in the direction orthogonal to it. A similar approach has been presented in [26]. Locally adaptive metric was proposed using a Chi-squared distance that measures the similarities between two instances in terms of the difference between their two class posterior probabilities. In [48], a distance metric has been presented by learning a linear transformation of the input space such that in the transformed space, k-NN performs well. A method for

learning a Mahalanobis distance measure by semi-definite linear programming has been proposed in [112]. The metric is trained with the goal that the k nearest neighbors always belong to the same class while instances from different classes are separated by a large margin. In [110], an adaptive distance metric has been proposed. It consists in normalizing the ordinary distance (e.g. Euclidean one) between the query instance $\mathbf{x}$ and a training instance $\mathbf{x}_i$ by the shortest distance between $\mathbf{x}_i$ and training instances belonging to classes different from the class of $\mathbf{x}_i$. Although these proposed metrics may improve the performance of the k-NN classifier, their computational complexity is higher than that of the conventional Euclidean distance.

We have to specify as well the value of the parameter k that controls the size of the neighborhood. A major issue in k-NN classification is how to find an optimal value of k. In general, the value of k depends on the size of the training data n. As shown in [17], k should vary with n in such a way that $k \longrightarrow \infty$ and $k/n \longrightarrow 0$ as $n \longrightarrow \infty$. However, for finite values of n, there is no theoretical guideline for choosing the value of k. In [85], a study on the relationship between the size of the training dataset and the parameter k, and the impact of k on classification accuracy have been reviewed. It has been shown that for large training sets, a broad set of values of k leads to similar results, while small training sets require more careful selection of k. For larger training sizes, accuracy becomes increasingly stable with respect to k. In general, larger values of k tend to produce smoother models and are less sensitive to label noise; however, they increase the computational burden and include further training instances in label estimation, so there is no *locality* in that estimation [17]. In [109], a method for neighborhood size selection based on the concept of statistical confidence has been proposed. In this approach, a defined criterion is used to determine the needed value of k. The number of nearest neighbors is dynamically adjusted until a satisfactory level of confidence is reached. In [113], it has been shown that the best value of k not only depends on the training dataset, but also on the given instance to classify. Instead of using a fixed value of k, a local value is estimated for each query instance. The adaptive choice of k has also been studied in [46]. Using different values of k instead of single value adds more flexibility to the classification process, but, however, makes it more difficult and more computationally complex because, the optimization of k has to be made for each instance. Cross-validation is still the most widely used approach to estimate the optimal value of the neighborhood parameter k.

2.4 Label correlation in multi-label applications

In multi-label learning, the possibility of joint membership of an instance to several classes may imply the existence of some information in the label space about the interdependency between different labels. The assignment of class ω to an instance $\mathbf{x}$ may provide information about the membership of that instance to other classes. Label correlation exists when, the possibility for an instance to belong to a class depends on its membership to other classes. For example, a document with the topic *politics* is unlikely to be labeled as *entertainment*, but the probability that the document belongs to class *economic* is high.

In general, relationships between labels have high order or even full order, i.e., there is a relation between a label and all remaining labels, but these relations are more difficult to represent than second-order relations, i.e., relations that exist between each pair of labels. Label correlation can be represented in the form of a contingency matrix *mat* that allows us to express only *second-order* relations between labels. Given a multi-labeled dataset $\mathcal{D}$ with Q possible labels, $mat[q][r] = \Pr(\mathrm{H}_1^q|\mathrm{H}_1^r)$, where q and $r \in \{1, \ldots, \mathrm{Q}\}$ with $q \neq r$, indicates the second-order relationship between labels ω_q and ω_r. $\Pr(\mathrm{H}_1^q|\mathrm{H}_1^r)$ represents the proportion of data in $\mathcal{D}$ that are assigned label ω_q, knowing that they also belong to ω_r. $mat[q][q] = \Pr(\mathrm{H}_1^q)$ indicates the frequency of label ω_q in the dataset $\mathcal{D}$. Figures 2.1, 2.2 and 2.3 show, respectively, the contingency matrices for the emotion ($\mathrm{Q} = 6$), scene ($\mathrm{Q} = 6$) and yeast ($\mathrm{Q} = 14$) datasets used in our experiments, which will be described in Chapter 5. For example, in the emotion dataset, each object represents a song and is labeled by the emotions evoked by this song. We can see in Figure 2.1 that $mat[1][4] = \Pr(\mathrm{H}_1^1|\mathrm{H}_1^4) = 0$, meaning that labels ω_1 and ω_4 cannot occur together. This is easily interpretable, as ω_1 corresponds to "amazed-surprised" while ω_4 corresponds to "quiet-still", and these two emotions are clearly opposite. We can also see that $mat[5][4] = \Pr(\mathrm{H}_1^5|\mathrm{H}_1^4) = 0.6$, which means that ω_5 representing "sad-lonely" frequently coexists in the label sets with ω_4. We can see from these examples that labels in multi-labeled datasets are often correlated, and exploiting relationships between labels will be very helpful for improving classification performance.

The most intuitive and straightforward way for multi-label learning is the BR approach, which decomposes a multi-label classification problem into several binary classification problems; one binary classifier is trained for each label and used to predict

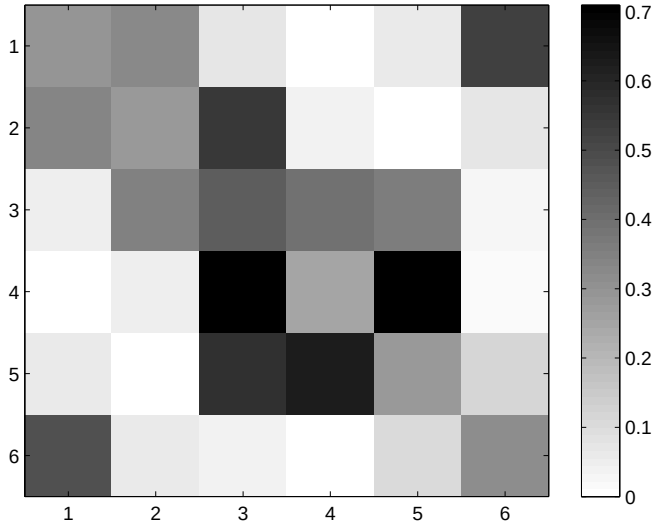

Figure 2.1: Contingency matrix of emotion dataset

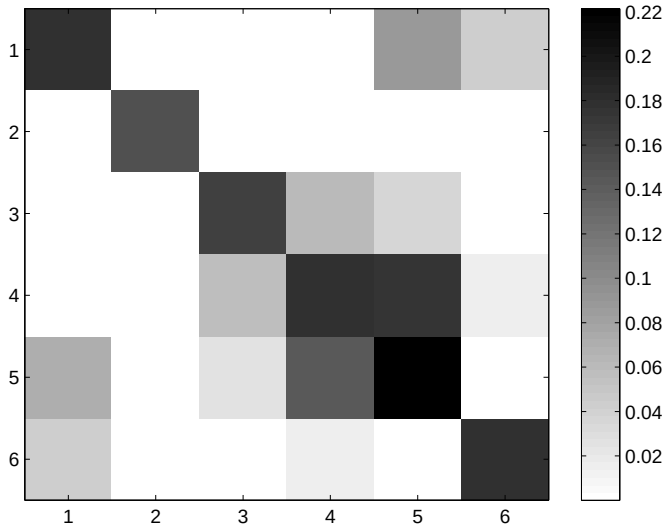

Figure 2.2: Contingency matrix of scene dataset

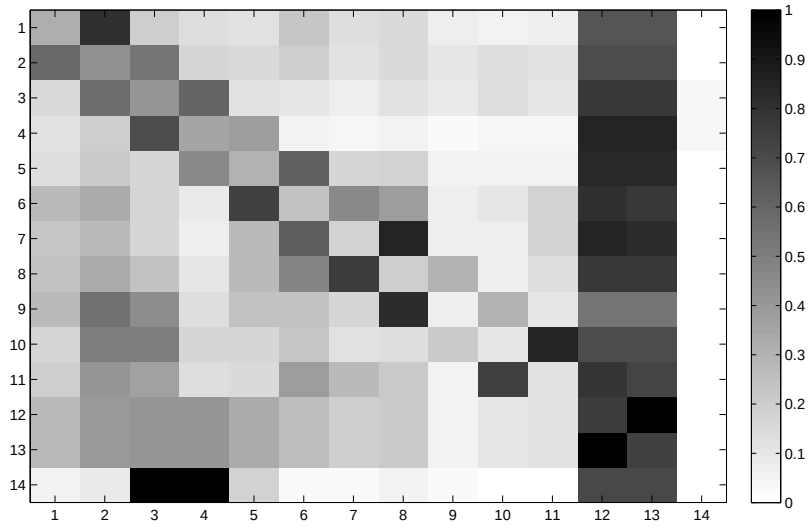

Figure 2.3: Contingency matrix of yeast dataset

whether, for a given test instance, this label is relevant or not (see Section 1.2.1). An advantage of the BR approach is that it is simple, intuitive, and a multi-label classifier can be built by directly using any state-of-the-art binary classification algorithm. However, BR learns the binary classifiers independently, and ignores any relation between labels. It may also predict labels that would never co-occur in reality.

In fact, for optimal performance, a probabilistic multi-label classifier should estimate the set of labels with the highest *joint* probability, instead of the combination of labels with largest individual probabilities. For example, given an event E about an instance to classify $\mathbf{x}$, we suppose that the joint posterior probability $\Pr(\mathrm{H}_b^1, \mathrm{H}_{b'}^2|\mathrm{E})$ of the two possible classes ω_1 and ω_2 are shown in Table 2.1. Suppose also that we trained a single-label probabilistic classifier for each of the two classes, and, as a result, we obtained the two individual posterior probabilities $\Pr(\mathrm{H}_b^1|\mathrm{E})$ and $\Pr(\mathrm{H}_{b'}^2|\mathrm{E})$. We can see that $\Pr(\mathrm{H}_0^1|\mathrm{E}) = 0.3$ is less than $\Pr(\mathrm{H}_1^1|\mathrm{E}) = 0.7$, and the binary approach will assign class ω_1 to instance $\mathbf{x}$. For the same reason, $\mathbf{x}$ is assigned class ω_2. However, if we take a look at Table 2.1, we can remark that $\Pr(\mathrm{H}_0^1, \mathrm{H}_1^2|\mathrm{E}) = 0.4$ is bigger than $\Pr(\mathrm{H}_1^1, \mathrm{H}_1^2|\mathrm{E}) = 0.3$, which means that the true label set of $\mathbf{x}$ only contains class ω_2. Therefore, combining independent binary classifiers may not be really effective for the

purpose of multi-label classification, as the mutual correlations among different classes are completely ignored. A more adequate approach for multi-label learning might be to take into account the different combinations of labels and to compute the corresponding joint probabilities. In practice, in the absence of prior knowledge, this approach estimates the joint probabilities from a given training dataset $\mathcal{D}$ of size n. Moreover, the number of label combinations expands exponentially with the increase of the number of possible labels Q, and the number of joint probabilities to be estimated is upper bounded by $\min(n, 2^{\mathrm{Q}})$. Consequently, estimating all joint probabilities has higher computational complexity but, some joint probabilities have to be estimated from possibly small number of training instances, which may introduce some bias in the learning process and degrade the overall classification accuracy. In fact, when calculating the joint probabilities of different possible label combinations, only a small number of training instances may be associated with each combination, specially if it contains many labels. Another limitation of this approach is that an instance to classify can only be associated to a label set that exists in the training dataset.

Table 2.1: An example of joint distributions of two labels.

$\Pr(\mathrm{H}_b^1, \mathrm{H}_{b'}^2 \mid \mathrm{E})$	$b=0$	$b=1$	$\Pr(\mathrm{H}_{b'}^2 \mid \mathrm{E})$
$b'=0$	0	**0.4**	0.4
$b'=1$	0.3	0.3	**0.6**
$\Pr(\mathrm{H}_b^1 \mid \mathrm{E})$	0.3	**0.7**	

In this chapter, we propose a Bayesian multi-label classification method based on the k-NN rule, which is able to capture correlations among labels while maintaining acceptable computational complexity. This method is called DMLkNN for dependent multi-label k-nearest neighbor. It is BR-based approach in the sense that a binary decision is made separately for each label given an instance to classify, but it overcomes the label independence assumption of BR. In our method, label correlation is exploited by extracting statistical information from training instances, which will be used to assign or not each label to a given test instance. This method is a generalization of the MLkNN algorithm proposed in [129]. In this algorithm, a decision is made separately for each label by taking into account the number of neighbors belonging at least to that label. Thus, this method fails to take into consideration the interdependency between labels. In contrast, after identifying the k-NNs of the instance to classify, our method

uses a MAP rule for each label, which takes into account the numbers of neighboring instances belonging to the different labels instead of only considering the number of neighbors having the label in question.

2.5 DMLkNN for multi-label classification

As in the first chapter, let $\mathbb{X} = \mathbb{R}^d$ denote the domain of instances, each one represented by a d-dimensional feature vector, and let $\mathcal{Y} = \{\omega_1, \omega_2, \ldots, \omega_{\mathrm{Q}}\}$ be the finite set of labels. Let $\mathcal{D} = \{(\mathbf{x}_1, Y_1), (\mathbf{x}_2, Y_2), \ldots, (\mathbf{x}_n, Y_n)\}$ represent the multi-labeled dataset, consisting of n training examples, independently drawn from $\mathbb{X} \times 2^{\mathcal{Y}}$, and identically distributed, where $\mathbf{x}_i \in \mathbb{X}$ and $Y_i \in 2^{\mathcal{Y}}$. The DMLkNN method learns a multi-label classifier $\mathcal{H} : \mathbb{X} \to 2^{\mathcal{Y}}$ from the given training data, which predicts a set of labels to each unseen instance $\mathbf{x} \in \mathbb{X}$. In addition to $\mathcal{H}$, DMLkNN defines a scoring function $f : \mathbb{X} \times \mathcal{Y} \to \mathbb{R}$ that assigns a real number to each instance/label combination. For each class $\omega \in \mathcal{Y}$, the score $f(\mathbf{x}, \omega)$ represents the probability that ω is relevant for the instance $\mathbf{x}$. The scoring function f is used to rank the labels corresponding to their relevance for the instance to classify. Note that the multi-label classifier $\mathcal{H}(\cdot)$ and the scoring function $f(\cdot, \cdot)$ are linked by the following relation:

$$\mathcal{H}(\mathbf{x}) = \{\omega \in \mathcal{Y} | f(\mathbf{x}, \omega) > t\},$$

where t is a threshold value.

Given an instance $\mathbf{x}$ and its associated label set $Y \subseteq \mathcal{Y}$, let $\mathcal{N}_{\mathbf{x}}^k$ denote the set of the k closest training examples of $\mathbf{x}$ in $\mathcal{D}$ according to a distance function d(.,.), and let $\mathbf{y}_{\mathbf{x}}$ be the Q-dimensional *category* vector of $\mathbf{x}$ whose qth component indicates if $\mathbf{x}$ belongs to class ω_q or not:

$$\mathbf{y}_{\mathbf{x}}(q) = \begin{cases} 1 & \text{if } \omega_q \in Y \\ 0 & \text{otherwise} \end{cases} \qquad \forall\ q \in \{1, \ldots, \mathrm{Q}\}.$$

Let us represent by $\mathbf{c}_{\mathbf{x}}$ the Q-dimensional *membership counting* vector of $\mathbf{x}$, the qth component of which indicates how many examples amongst the k-NNs of $\mathbf{x}$ belong to class ω_q:

$$\mathbf{c}_{\mathbf{x}}(q) = \sum_{\mathbf{x}_i \in \mathcal{N}_{\mathbf{x}}^k} \mathbf{y}_{\mathbf{x}_i}(q), \qquad \forall\ q \in \{1, \ldots, \mathrm{Q}\}.$$

2.5.1 MAP principle

Let $\mathbf{x}$ now denote an instance to classify. Like in all k-NN based methods, for the test instance $\mathbf{x}$, the set $\mathcal{N}_{\mathbf{x}}^{k}$ of its k nearest neighbors should be firstly identified. Under the multi-label assumption, the counting vector $\mathbf{c}_{\mathbf{x}}$ is computed. As mentioned before, let H_1^q denote the hypothesis that $\mathbf{x}$ belongs to class ω_q, and H_0^q the hypothesis that $\mathbf{x}$ should not be assigned ω_q. Let E_j^q ($j \in \{0, 1, \ldots, k\}$) denote the event that there are exactly j instances in $\mathcal{N}_{\mathbf{x}}^{k}$ belonging to class ω_q. To determine the qth component of the category vector $\mathbf{y}_{\mathbf{x}}$ for instance $\mathbf{x}$, the MLkNN algorithm uses the following MAP [129]:

$$\widehat{\mathbf{y}}'_{\mathbf{x}}(q) = \arg\max_{b\in\{0,1\}} \Pr(\mathrm{H}_b^q|\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}), \tag{2.1}$$

while for the DMLkNN algorithm, the following MAP is used:

$$\begin{aligned}\widehat{\mathbf{y}}_{\mathbf{x}}(q) &= \arg\max_{b\in\{0,1\}} \Pr(\mathrm{H}_b^q|\bigwedge_{\omega_l\in\mathcal{Y}} \mathrm{E}^l_{\mathbf{c}_{\mathbf{x}}(l)}) \\ &= \arg\max_{b\in\{0,1\}} \Pr(\mathrm{H}_b^q|\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l\in\mathcal{Y}\backslash\{\omega_q\}} \mathrm{E}^l_{\mathbf{c}_{\mathbf{x}}(l)}).\end{aligned} \tag{2.2}$$

In contrast to decision rule (2.1), we can see from Equation (2.2) that the assignment of label ω_q to the test instance $\mathbf{x}$ depends not only on the event that there are exactly $\mathbf{c}_{\mathbf{x}}(q)$ instances having label ω_q in $\mathcal{N}_{\mathbf{x}}^{k}$, i.e., $\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}$, but also on $\bigwedge_{\omega_l\in\mathcal{Y}\backslash\{\omega_q\}} \mathrm{E}^l_{\mathbf{c}_{\mathbf{x}}(l)}$, which is the event that there are exactly $\mathbf{c}_{\mathbf{x}}(l)$ instances having label ω_l in $\mathcal{N}_{\mathbf{x}}^{k}$, for each $\omega_l \in \mathcal{Y}\backslash\{\omega_q\}$. Thus, it is clear that label correlation is taken into account in (2.2) since all the components of the counting vector $\mathbf{c}_{\mathbf{x}}$ are involved in the assignment or not of label ω_q to $\mathbf{x}$, which is not the case in Equation (2.1).

2.5.2 Posterior probability estimation

Regarding the counter vector $\mathbf{c}_{\mathbf{x}}$, the number of possible events $\bigwedge_{\omega_l\in\mathcal{Y}} \mathrm{E}^l_{\mathbf{c}_{\mathbf{x}}(l)}$ is upper bounded by k^{Q}. This means that, in addition to the complexity problem, the estimation of (2.2) from a relatively small training set will not be accurate. To overcome this difficulty, we will adopt a fuzzy approximation for (2.2). This approximation is based on the event F_j^l, $j \in \{0, 1, \ldots, k\}$, which is the event that there are *approximately* j instances in $\mathcal{N}_{\mathbf{x}}^{k}$ belonging to class ω_l, i.e., F_j^l, denotes the event that the number of instances in $\mathcal{N}_{\mathbf{x}}^{k}$ that are assigned label ω_l is in the interval $[j-\delta; j+\delta]$, where

$\delta \in \{0, \ldots, k\}$ is a *fuzziness* parameter. As a consequence, we can derive a fuzzy MAP rule:

$$\widehat{\mathbf{y}}_{\mathbf{x}}(q) = \underset{b\in\{0,1\}}{\arg\max} \Pr(\mathrm{H}_b^q | \bigwedge_{\omega_l \in \mathcal{Y}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)}). \tag{2.3}$$

To remain closer to the initial formulation and for comparison with MLkNN, (2.3) will be replaced by the following rule:

$$\widehat{\mathbf{y}}_{\mathbf{x}}(q) = \underset{b\in\{0,1\}}{\arg\max} \Pr(\mathrm{H}_b^q | \mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l \in \mathcal{Y}\setminus\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)}). \tag{2.4}$$

For large values of δ, the results of our method will be similar to those of MLkNN. In fact, for $\delta = k$, the MLkNN algorithm is a particular case of the DMLkNN algorithm, where $\bigwedge_{\omega_l \in \mathcal{Y}\setminus\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)}$ will be *certain* event because for each $\omega_l \in \mathcal{Y}\setminus\{\omega_q\}$, the number of instances in $\mathcal{N}^k_{\mathbf{x}}$ belonging to class ω_l will surely be in the interval $[j-k; j+k]$. For small values of δ, the assignment or not of label ω_q to test instance $\mathbf{x}$ will not only depend on the number of instances in $\mathcal{N}^k_{\mathbf{x}}$ that belong to label ω_q, but also on the number of instances in $\mathcal{N}^k_{\mathbf{x}}$ belonging to the remaining labels.

Using the Bayes' rule, Equations (2.1) and (2.4) can be written as follows:

$$\begin{aligned} \widehat{\mathbf{y}}'_{\mathbf{x}}(q) &= \underset{b\in\{0,1\}}{\arg\max} \frac{\Pr(\mathrm{H}_b^q)\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}|\mathrm{H}_b^q)}{\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)})} \\ &= \underset{b\in\{0,1\}}{\arg\max} \Pr(\mathrm{H}_b^q)\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}|\mathrm{H}_b^q). \end{aligned} \tag{2.5}$$

$$\begin{aligned} \widehat{\mathbf{y}}_{\mathbf{x}}(q) &= \underset{b\in\{0,1\}}{\arg\max} \frac{\Pr(\mathrm{H}_b^q)\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l \in \mathcal{Y}\setminus\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)}|\mathrm{H}_b^q)}{\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l \in \mathcal{Y}\setminus\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)})} \\ &= \underset{b\in\{0,1\}}{\arg\max} \Pr(\mathrm{H}_b^q)\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l \in \mathcal{Y}\setminus\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)}|\mathrm{H}_b^q). \end{aligned} \tag{2.6}$$

To rank labels in $\mathcal{Y}$, a Q-dimensional real-valued vector $\mathbf{r}_{\mathbf{x}}$ can be calculated. The

qth component of $\mathbf{r}_{\mathbf{x}}$ is defined as the posterior probability $\Pr(\mathrm{H}_1^q|\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l\in\mathcal{Y}\setminus\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)})$:

$$\begin{aligned}
\mathbf{r}_{\mathbf{x}}(q) &= \Pr(\mathrm{H}_1^q|\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l\in\mathcal{Y}\setminus\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)}) \\
&= \frac{\Pr(\mathrm{H}_1^q)\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l\in\mathcal{Y}\setminus\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)}|\mathrm{H}_1^q)}{\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l\in\mathcal{Y}\setminus\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)})} \\
&= \frac{\Pr(\mathrm{H}_1^q)\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l\in\mathcal{Y}\setminus\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)}|\mathrm{H}_1^q)}{\sum_{b\in\{0,1\}}\Pr(\mathrm{H}_b^q)\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l\in\mathcal{Y}\setminus\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)}|\mathrm{H}_b^l)}.
\end{aligned} \tag{2.7}$$

For comparison, the real-valued vector $\mathbf{r}'_{\mathbf{x}}$ for MLkNN has the following expression:

$$\begin{aligned}
\mathbf{r}'_{\mathbf{x}}(q) &= \Pr(\mathrm{H}_1^q|\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}) \\
&= \frac{\Pr(\mathrm{H}_1^q)\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}|\mathrm{H}_1^q)}{\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)})} \\
&= \frac{\Pr(\mathrm{H}_1^q)\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}|\mathrm{H}_1^q)}{\sum_{b\in\{0,1\}}\Pr(\mathrm{H}_b^q)\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}|\mathrm{H}_b^l)}.
\end{aligned} \tag{2.8}$$

In order to determine the category vector $\widehat{\mathbf{y}}_{\mathbf{x}}$ and the real-valued vector $\mathbf{r}_{\mathbf{x}}$ of instance $\mathbf{x}$, we need to determine the prior probabilities $\Pr(\mathrm{H}_b^l)$ and the likelihoods $\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l\in\mathcal{Y}\setminus\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)}|\mathrm{H}_b^q)$, for each $q \in \{1\cdots\mathrm{Q}\}$, and $b \in \{0,1\}$. These probabilities are estimated from a training dataset $\mathcal{D}$.

Given an instance $\mathbf{x}$ to classify, the output of the DMLkNN method for multi-classification is determined as follows:

$$\mathcal{H}(\mathbf{x}) = \{\omega_q \in \mathcal{Y} |\ \widehat{\mathbf{y}}_{\mathbf{x}}(q) = 1\},$$

and

$$f(\mathbf{x}, \omega_q) = \mathbf{r}_{\mathbf{x}}(q),\ \text{for each } \omega_q \in \mathcal{Y}.$$

Figure 2.4 shows the pseudo code of the DMLkNN algorithm. The value of δ may be selected through cross-validation and provided as input to the algorithm. The prior probabilities $\Pr(\mathrm{H}_b^q)$, $b = \{0,1\}$, for each class ω_q are first calculated and the number of instances belonging to each label is counted (steps 1 to 3):

$$\begin{cases} \Pr(\mathrm{H}_1^q) &= \frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_{\mathbf{x}_i}(q) \\ \Pr(\mathrm{H}_0^q) &= 1 - \Pr(\mathrm{H}_1^q). \end{cases} \tag{2.9}$$

$[\mathbf{y}_{\mathbf{x}}, \mathbf{r}_{\mathbf{x}}] = \mathrm{DML}k\mathrm{NN}(\mathcal{D}, \mathbf{x}, k, s, \delta)$

%Computing the prior probabilities and the number of instances belonging to each class

1. **For** $q = 1 \cdots \mathrm{Q}$
2. $\Pr(\mathrm{H}_1^q) = (\sum_{i=1}^{m} \mathbf{y}_{\mathbf{x}_i}(q))/(n)$; $\Pr(\mathrm{H}_0^q) = 1 - \Pr(\mathrm{H}_1^q)$;
3. $\mathbf{u}(q) = \sum_{i=1}^{n} \mathbf{y}_{\mathbf{x}_i}(q)$; $\mathbf{u}'(q) = n - \mathbf{u}(q)$;
   EndFor
   %For each test instance $\mathbf{x}$
4. **Identify** $N(\mathbf{x})$ and $\mathbf{c}_{\mathbf{x}}$
   %Counting the training instances whose membership counting vectors satisfy the constraints (2.11)
5. **For** $q = 1 \cdots \mathrm{Q}$
6. $\mathbf{v}(q) = 0$; $\mathbf{v}'(q) = 0$
   EndFor
7. **For** $i = 1 \cdots n$
8. **Identify** $N(\mathbf{x}_i)$ and $\mathbf{c}_{\mathbf{x}_i}$
9. **If** $\mathbf{c}_{\mathbf{x}}(q) - \delta \leq \mathbf{c}_{\mathbf{x}_i}(q) \leq \mathbf{c}_{\mathbf{x}}(q) + \delta, \forall\, q \in \mathcal{Y}$ **Then**
10. **For** $q = 1 \cdots \mathrm{Q}$
11. **If** $\mathbf{c}_{\mathbf{x}_i}(q) == \mathbf{c}_{\mathbf{x}}(q)$ **Then**
12. **If** $\mathbf{y}_{\mathbf{x}_i}(q) == 1$ **Then** $\mathbf{v}(q) = \mathbf{v}(q) + 1$;
    Else $\mathbf{v}'(q) = \mathbf{v}'(q) + 1$;
    EndFor
    EndFor
    %Computing $\mathbf{y}_{\mathbf{x}}$ and $\mathbf{r}_{\mathbf{x}}$
13. **For** $q = 1 \cdots \mathrm{Q}$
14. $\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l \in \mathcal{Y} \setminus \{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)} | \mathrm{H}_1^q) = (s + \mathbf{v}(q))/(s \times \mathrm{Q} + \mathbf{u}(q))$;
15. $\Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l \in \mathcal{Y} \setminus \{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)} | \mathrm{H}_0^q) = (s + \mathbf{v}'(q))/(s \times \mathrm{Q} + \mathbf{u}'(q))$;
16. $\mathbf{y}_{\mathbf{x}}(q) = \arg\max_{b \in \{0,1\}} \Pr(\mathrm{H}_b^q) \Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l \in \mathcal{Y} \setminus \{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)} | \mathrm{H}_b^q)$
17. $\mathbf{r}_{\mathbf{x}}(q) = \dfrac{\Pr(\mathrm{H}_1^q) \Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l \in \mathcal{Y} \setminus \{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)} | \mathrm{H}_1^q)}{\sum_{b \in \{0,1\}} \Pr(\mathrm{H}_b^q) \Pr(\mathrm{E}^q_{\mathbf{c}_{\mathbf{x}}(q)}, \bigwedge_{\omega_l \in \mathcal{Y} \setminus \{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_{\mathbf{x}}(l)} | \mathrm{H}_b^q)}$

EndFor

Figure 2.4: DMLkNN algorithm.

Recall that n is the number of training instances. $\mathbf{u}(q)$ counts the number of instances belonging to class ω_q, and $\mathbf{u}'(q)$ indicates the number of instances not having ω_q in their label sets:

$$\left\{ \begin{array}{rcl} \mathbf{u}(q) & = & \sum_{i=1}^{n} \mathbf{y}_{\mathbf{x}_i}(q) \\ \mathbf{u}'(q) & = & n - \mathbf{u}(q). \end{array} \right. \tag{2.10}$$

For test instance $\mathbf{x}$, the k-NNs are identified and the membership counting vector $\mathbf{c}_\mathbf{x}$ is determined (step 4). In order to assign or not label ω_q to $\mathbf{x}$, we must determine the likelihoods $\Pr(\mathrm{E}^q_{\mathbf{c}_\mathbf{x}(q)}, \bigwedge_{\omega_l \in \mathcal{Y}\backslash\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_\mathbf{x}(l)} | \mathrm{H}^q_b)$, $b \in \{0,1\}$, using the training instances such as their corresponding membership counting vectors satisfy the following constraints:

$$\left\{ \begin{array}{l} \mathbf{c}_{\mathbf{x}_i}(q) = \mathbf{c}_\mathbf{x}(q) \\ \mathbf{c}_\mathbf{x}(l) - \delta \leq \mathbf{c}_{\mathbf{x}_i}(l) \leq \mathbf{c}_\mathbf{x}(l) + \delta, \quad \text{for each } \omega_l \in \mathcal{Y}\backslash\{\omega_q\}. \end{array} \right. \tag{2.11}$$

This is illustrated in steps 5 to 12. The number of instances from the training set verifying these constraints, and belonging to class ω_q is stored in $\mathbf{v}(q)$. The number of remaining instances verifying the previous constraints and not having ω_q in their sets of labels is stored in $\mathbf{v}'(q)$. The likelihoods $\Pr(\mathrm{E}^q_{\mathbf{c}_\mathbf{x}(q)}, \bigwedge_{\omega_l \in \mathcal{Y}\backslash\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_\mathbf{x}(l)} | \mathrm{H}^q_b)$, $b \in \{0,1\}$, are then computed:

$$\left\{ \begin{array}{rcl} \Pr(\mathrm{E}^q_{\mathbf{c}_\mathbf{x}(q)}, \bigwedge_{\omega_l \in \mathcal{Y}\backslash\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_\mathbf{x}(l)} | \mathrm{H}^q_1) & = & \frac{s+\mathbf{v}(l)}{s\times \mathrm{Q}+\mathbf{u}(l)} \\ \Pr(\mathrm{E}^q_{\mathbf{c}_\mathbf{x}(q)}, \bigwedge_{\omega_l \in \mathcal{Y}\backslash\{\omega_q\}} \mathrm{F}^l_{\mathbf{c}_\mathbf{x}(l)} | \mathrm{H}^q_0) & = & \frac{s+\mathbf{v}'(l)}{s\times \mathrm{Q}+\mathbf{u}'(l)}, \end{array} \right. \tag{2.12}$$

where s is a smoothing parameter [86]. Smoothing is commonly used to avoid zero probability estimates. When $s = 1$, it is called Laplace smoothing. Finally, the category vector $\mathbf{y}_\mathbf{x}$ and the real-valued vector $\mathbf{r}_\mathbf{x}$ to rank labels in $\mathcal{Y}$ are calculated using equations (2.6) and (2.8), respectively (steps 13 to 17).

Note that, in the MLkNN algorithm, only the first constraint in (2.11) is considered in order to compute the likelihoods $\Pr(\mathrm{E}^q_{\mathbf{c}_\mathbf{x}(q)} | \mathrm{H}^q_b)$, $b \in \{0,1\}$. As a result, the number of examples in the learning set satisfying this contraint is larger than the number of examples satisfying (2.11). Thus, the MLkNN and DMLkNN should not necessary be compared with the same smoothing parameter.

2.6 Illustration on a simulated dataset

In this section, we illustrate the behavior of the DMLkNN and MLkNN methods using simulated data.

The simulated dataset contains 1019 instances in $\mathbb{R}^2$ belonging to three possible classes, $\mathcal{Y} = \{\omega_1, \omega_2, \omega_3\}$. The data were generated from seven Gaussian distributions with means (0,0), (1,0), (0.5,0), (0.5,1), (0.25,0.6), (0.75,0.6), (0.5,0.5) , respectively, and equal covariance matrix $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. The number of instances in each class is chosen arbitrarily (see Table 2.2). Taking into account the geometric distribution of the gaussian data, the instances of each set were respectively assigned to label(s) $\{\omega_1\}$, $\{\omega_2\}$, $\{\omega_1, \omega_2\}$, $\{\omega_3\}$, $\{\omega_1, \omega_3\}$, $\{\omega_2, \omega_3\}$, $\{\omega_1, \omega_2, \omega_3\}$.

Table 2.2: Summary of the simulated data set.

Label set	Number of instances
$\{\omega_1\}$	150
$\{\omega_2\}$	162
$\{\omega_1, \omega_2\}$	304
$\{\omega_3\}$	262
$\{\omega_1, \omega_3\}$	43
$\{\omega_2, \omega_3\}$	78
$\{\omega_1, \omega_2, \omega_3\}$	20

Figure 2.5 shows the neighboring training instances and the estimated label set for a test instance $\mathbf{x}$ using DMLkNN and MLkNN. For both methods, k was set to 8, and Laplace smoothing ($s = 1$) was used. For DMLkNN, δ was fixed to 1. Hereafter, for the test instance in question, we will describe the different steps for the estimation of the label set of $\mathbf{x}$ using the DMLkNN and MLkNN algorithms. For the sake of clarity, we will recall the definition of some events introduced before. The membership counting vector of the test instance is $\mathbf{c}_\mathbf{x} = (7, 3, 2)$. Using the DML$k$NN method, in order to estimate the label set of $\mathbf{x}$, the following probabilities have to be computed from Equation (2.6):

$$\begin{aligned}
\widehat{\mathbf{y}}_\mathbf{x}(1) &= \underset{b\in\{0,1\}}{\arg\max}\ \Pr(\mathrm{H}_b^1)\Pr(\mathrm{E}_7^1, \mathrm{F}_3^2, \mathrm{F}_2^3|\mathrm{H}_b^1) \\
\widehat{\mathbf{y}}_\mathbf{x}(2) &= \underset{b\in\{0,1\}}{\arg\max}\ \Pr(\mathrm{H}_b^2)\Pr(\mathrm{E}_3^2, \mathrm{F}_7^1, \mathrm{F}_2^3|\mathrm{H}_b^2) \\
\widehat{\mathbf{y}}_\mathbf{x}(3) &= \underset{b\in\{0,1\}}{\arg\max}\ \Pr(\mathrm{H}_b^3)\Pr(\mathrm{E}_2^3, \mathrm{F}_7^1, \mathrm{F}_3^2|\mathrm{H}_b^3).
\end{aligned}$$

We recall that E_7^1 is the event that there are seven instances in $\mathcal{N}_\mathbf{x}^k$ which have label ω_1, F_3^2 is the event that the number of instances in $\mathcal{N}_\mathbf{x}^k$ belonging to label ω_2 is in the

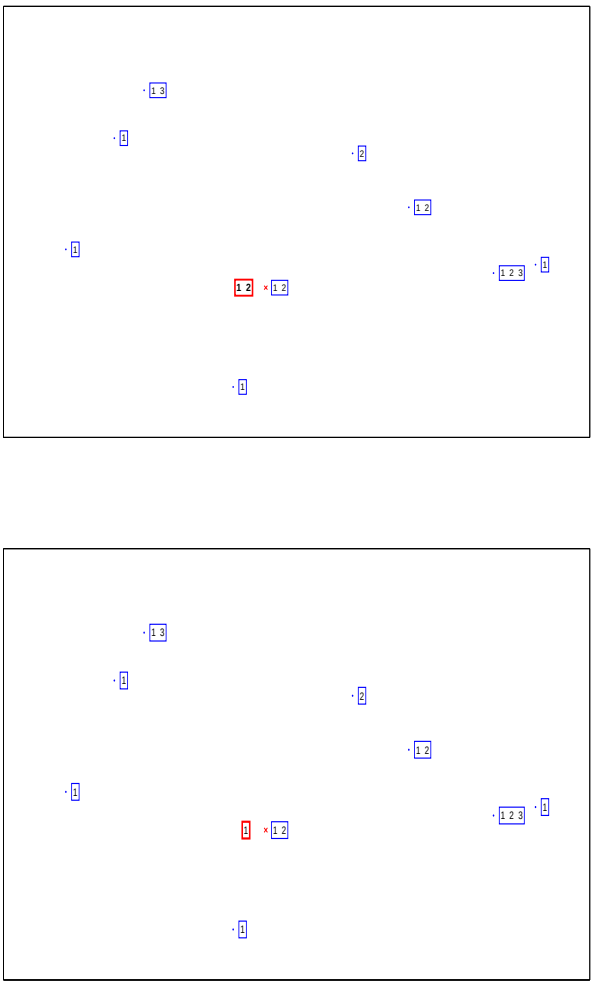

Figure 2.5: Estimated label set (in bold) for a test instance using the DMLkNN (top) and MLkNN (bottom) methods.

interval $[3-\delta; 3+\delta] = [2, 4]$. In contrast, for estimating the label set of the unseen instance using the MLkNN method, the following probabilities have to be computed

from Equation (2.5):

$$\begin{aligned}
\widehat{\mathbf{y}}'_{\mathbf{x}}(1) &= \arg\max_{b\in\{0,1\}} \Pr(\mathrm{H}_b^1)\Pr(\mathrm{E}_7^1|\mathrm{H}_b^1) \\
\widehat{\mathbf{y}}'_{\mathbf{x}}(2) &= \arg\max_{b\in\{0,1\}} \Pr(\mathrm{H}_b^2)\Pr(\mathrm{E}_3^2|\mathrm{H}_b^2) \\
\widehat{\mathbf{y}}'_{\mathbf{x}}(3) &= \arg\max_{b\in\{0,1\}} \Pr(\mathrm{H}_b^3)\Pr(\mathrm{E}_2^3|\mathrm{H}_b^3).
\end{aligned}$$

First, the prior probabilities are computed from the training set according to Equation (2.9):

$$\begin{aligned}
&\Pr(\mathrm{H}_1^1) = 0.4527 \qquad \Pr(\mathrm{H}_0^1) = 0.5473 \\
&\Pr(\mathrm{H}_1^2) = 0.5038 \qquad \Pr(\mathrm{H}_0^2) = 0.4962 \\
&\Pr(\mathrm{H}_1^3) = 0.4396 \qquad \Pr(\mathrm{H}_0^3) = 0.5604.
\end{aligned}$$

Second, the posterior probabilities for the DMLkNN and MLkNN algorithms are calculated [1] using the training set:

$$\begin{aligned}
\Pr(\mathrm{E}_7^1, \mathrm{F}_3^2, \mathrm{F}_2^3|\mathrm{H}_1^1) = 0.0478 \qquad \Pr(\mathrm{E}_7^1, \mathrm{F}_3^2, \mathrm{F}_2^3|\mathrm{H}_0^1) = 0.0139 \\
\Pr(\mathrm{E}_3^2, \mathrm{F}_7^1, \mathrm{F}_2^3|\mathrm{H}_1^2) = 0.0237 \qquad \Pr(\mathrm{E}_3^2, \mathrm{F}_7^1, \mathrm{F}_2^3|\mathrm{H}_0^2) = 0.0218 \\
\Pr(\mathrm{E}_2^3, \mathrm{F}_7^1, \mathrm{F}_3^2|\mathrm{H}_1^3) = 0.0394 \qquad \Pr(\mathrm{E}_2^3, \mathrm{F}_7^1, \mathrm{F}_3^2|\mathrm{H}_0^3) = 0.1161 \\
\Pr(\mathrm{E}_7^1|\mathrm{H}_1^1) = 0.1108 \qquad \Pr(\mathrm{E}_7^1|\mathrm{H}_0^1) = 0.0431 \\
\Pr(\mathrm{E}_3^2|\mathrm{H}_1^2) = 0.1231 \qquad \Pr(\mathrm{E}_3^2|\mathrm{H}_0^2) = 0.1746 \\
\Pr(\mathrm{E}_2^3|\mathrm{H}_1^3) = 0.0655 \qquad \Pr(\mathrm{E}_2^3|\mathrm{H}_0^3) = 0.0593.
\end{aligned}$$

Using the prior and the posterior probabilities, the category vectors associated to the test instance by the DMLkNN and MLkNN algorithms can be calculated:

$$\begin{aligned}
&\widehat{\mathbf{y}}_{\mathbf{x}}(1) = 1 \qquad \widehat{\mathbf{y}}'_{\mathbf{x}}(1) = 1 \\
&\widehat{\mathbf{y}}_{\mathbf{x}}(2) = 1 \qquad \widehat{\mathbf{y}}'_{\mathbf{x}}(2) = 0 \\
&\widehat{\mathbf{y}}_{\mathbf{x}}(3) = 0 \qquad \widehat{\mathbf{y}}'_{\mathbf{x}}(3) = 0.
\end{aligned}$$

Thus, the estimated label set for test instance $\mathbf{x}$ given by the DMLkNN method is $\widehat{Y} = \{\omega_1, \omega_2\}$, while that given by ML$k$NN is $\widehat{Y}' = \{\omega_1\}$. The true label set for $\mathbf{x}$ is

[1] Using the DMLkNN method, this is done according to steps 7 to 15, as shown in Figure 2.4 and explained in Section 2.5.

$Y = \{\omega_1, \omega_2\}$. In this case, we can see that no error has occurred when estimating the label set of $\mathbf{x}$ using the DMLkNN method, while for the other method, the estimated label set is not identical to the ground truth label set. Seven training instances in $\mathcal{N}_{\mathbf{x}}^{k}$ have label ω_1 in their label sets while only three instances belong to label ω_2. In fact, the existence of label ω_1 in the neighborhood of $\mathbf{x}$ gives some information about the existence or not of label ω_2 in the label set of $\mathbf{x}$. If we take a look at the training dataset, we can remark that 14,7% of instances belong to ω_1, 15.9% to ω_2, and 29.8% to ω_1 and ω_2 simultaneously. Thus, the probability that an instance belongs to both classes ω_1 and ω_2 is approximately twice the probability that it belongs to only one of the two classes. DMLkNN is able to capture the relationship between labels ω_1 and ω_2 in order to improve the estimation of label sets, while MLkNN is not able to capture this correlation. This example shows that the DMLkNN method, which takes into account correlation between labels when calculating the assignment or not of a label to the test instance, may improve classification performance.

2.7 Conclusion

In this chapter, we have presented an original multi-label learning algorithm derived from the k-NN rule, in which the dependencies between labels are taken into account. Our method is based on the binary relevance approach, which is often criticized for its ignorance of correlation between labels. However, here, this disadvantage is overcome. The classification of an instance is carried out through local statistical information extracted from the k nearest neighbors of the instance to classify and using Bayesian inference. This method, called DMLkNN, generalizes the MLkNN algorithm presented in [129].

The illustrative example using a simulated dataset demonstrates the efficiency and the usefulness of our approach to represent and explore interdependencies between labels. However, for DMLkNN, as compared to MLkNN, there is one additional parameter that needs to be optimized, namely the fuzziness parameter δ. Moreover, MLkNN is faster than DMLkNN. In fact, in the MLkNN method, the likelihoods $\Pr(\mathrm{E}^{q}_{\mathbf{c_x}(q)}|\mathrm{H}^{q}_{b})$, $b \in \{0,1\}$, are calculated from the training set, stored and then just used when predicting the label set of each query instance. In contrast, using DMLkNN, the number of likelihoods $\Pr(\mathrm{E}^{q}_{\mathbf{c_x}(q)}, \bigwedge_{\omega_l \in \mathcal{Y}\setminus\{\omega_q\}} \mathrm{F}^{l}_{\mathbf{c_x}(l)}|\mathrm{H}^{q}_{b})$, $b \in \{0,1\}$, is much bigger, and thus, it will

not be an easy task to calculate these probabilities in advance and store them as in MLkNN. The probabilities are computed locally for each query instance.

Chapter 3

Multi-label learning under veristic variables

Summary

Veristic variables are fuzzy set-valued variables that can assume simultaneously multiple values with different degrees. In multi-label learning, class labels can be considered as veristic variables since each instance can belong to more than one class at the same time. Based on the approximate reasoning framework for representing and manipulating knowledge involving veristic variables, we propose in this chapter a veristic k-nearest neighbor rule for multi-label classification. The labeling of each instance is represented by two distributions: a first distribution called *Verity* which gives positive information about the labeling of this instance, and a second distribution called *Rebuff* which represents negative information about the different possible classes. Given an instance to classify, each neighbor represents a piece of knowledge about the labeling of this instance. The verity and rebuff distributions of the neighboring examples are discounted depending on the distance to the instance to classify and are then combined in order to determine the set of labels of that instance. This method is especially addressed to handle data with imprecise labels.

Résumé

Les variables *véristiques* sont des variables multi-valuées floues qui peuvent avoir plusieurs valeurs simultanément, mais avec différents degrés. Dans l'apprentissage multi-

label, les étiquettes des différents individus peuvent être considérés comme des variables véristiques, vu que chaque individu appartient à une ou plusieurs classes simultanément. Dans ce chapitre, nous proposons une méthode de classification multi-label basée sur la règle des k-plus proches voisins et utilisant le cadre de raisonnement approximatif des variables véristiques qui nous permet de représenter et manipuler de connaissances impliquant de telles variables. L'étiquetage de chaque exemple est représenté par deux distributions : une distribution appelée *Verity* qui donne des informations positives sur l'étiquetage de cet exemple, et une autre distribution appelée *Rebuff* représentant des informations négatives sur l'appartenance aux différentes classes possibles. Étant donné un nouveau individu à classifier, chaque voisin fournit une certaine connaissance sur l'étiquetage de cet individu. En tenant compte de la distance par rapport à l'individu à classifier, les distributions représentant l'étiquetage des différents voisins sont constituées en premier lieu, et sont ensuite combinées afin de déterminer l'ensemble de classes de cet individu. Cette méthode s'adresse spécialement à la classification de données étiquetées d'une façon imprécise.

3.1 Introduction

One may differentiate between two important classes of variables: *single-valued* and *set-valued* variables. Single-valued variables, also called *disjunctive* variables, are restricted to take one and only value in their universe of discourse. In contrast, set-valued variables, also called *conjunctive* ones, are allowed to take more than one value in their universe [125][32]. For instance, variables such as the current temperature, your day of birth are single-valued variables while, for example, the languages you speak, the countries you have visited are set-valued variables. When talking about fuzzy variables, single-valued ones are called *possibilistic* variables, while set-valued ones are called *veristic* variables [115].

In [119], an approximate reasoning framework has been proposed for the representation and manipulation of knowledge concerning veristic variables. Due to the fact that knowledge about set-valued variables may be uncertain and imprecise, the developed theory is based on fuzzy sets rather than crisp sets, in order to make it rich enough to handle all kinds of information.

As stated in the previous chapters, in multi-label learning problems, each instance may belong simultaneously to several classes, contrary to standard single-label problems where objects belong to only one class. Thus, in multi-label learning, the class label of each instance can be considered as a veristic variable. In this work, we propose a veristic k-nearest neighbor rule(k-NN) for multi-label learning. This method uses the approximate reasoning framework based on veristic variables for representing and combining knowledge about an unseen instance and predicting the corresponding set of labels. The labeling of each instance is represented by two distributions: a *verity* distribution that provides positive information about the labeling of this instance, and a *rebuff* distribution that represents negative informations about the possible classes. Given an unseen instance, each neighbor provides positive and negative information about the label set of this object according to the distance between the two patterns. The verity and rebuff distributions induced by each neighboring instance are discounted depending on the distance and are combined in order to determine the labeling of the instance to be classified.

This chapter is organized as follow. Section 3.2 presents the background on fuzzy sets and possibility theory. Elementary definitions and properties of fuzzy set and

possibility theories will be first recalled. The approximate reasoning framework for veristic variables will then be presented in Section 3.3. The representation of knowledge about veristic variables, verity and rebuff distributions, as well as the combination and discounting of veristic information will be addressed in this section. In Section 3.4, the task of multi-label learning in the framework of veristic variables will be studied. We will first discuss the labeling issue of multi-labeled instance in this framework, and the veristic-based method for multi-label classification will then be introduced. Finally, Section 3.5 will conclude this chapter.

3.2 Background

3.2.1 Fuzzy sets

In this section, we recall the basics of the theory of fuzzy sets. More details can be found in [123] and [31].

Let A be a (fuzzy or crisp) subset of the universe of discourse Ω. If A is a crisp set, each element ω in Ω is either a *full member* of A or not. In contrast, if A represents a fuzzy set, full membership is not necessary, and an element ω in Ω can be a member *to some degree*.

Given a fuzzy set A defined over Ω, a real value in the interval $[0, 1]$, represented by $A(\omega)$, is associated to each element $\omega \in \Omega$. $A(\omega)$ represents the *degree of membership* of ω in A, and the function $\omega \longrightarrow A(\omega)$ (sometimes denoted as μ_A) is referred to as the membership function of fuzzy set A. The concept of fuzzy set thus generalizes that of crisp set. In fact, the degree of membership of each element $\omega \in \Omega$ to a crisp set A of Ω takes values in $\{0, 1\}$ instead of the unit interval. Hereafter, we will review some definitions and properties concerning fuzzy sets.

3.2.1.1 Basic definitions

Two fuzzy subsets A and B are equal, if and only if the degree of membership is the same for each ω in Ω:

$$A = B \;\Leftrightarrow\; A(\omega) = B(\omega), \quad \forall\, \omega \in \Omega.$$

A is a subset of B if and only if, for all ω in Ω, the degree of membership of ω in A is less than the degree of membership of ω in B:

$$A \subseteq B \;\Leftrightarrow\; A(\omega) \leq B(\omega), \quad \forall\, \omega \in \Omega.$$

The complement of A, denoted by $\overline{A}$, is a fuzzy set of Ω for which the degree of membership of each element $\omega \in \Omega$ is defined as:

$$\overline{A}(\omega) = 1 - A(\omega), \quad \forall\, \omega \in \Omega.$$

The cardinality $|.|$ of a fuzzy set A may be defined as:

$$|A| = \sum_{\omega \in \Omega} A(\omega).$$

The α-cut of A, denoted by A_α, with $\alpha \in [0, 1]$, is defined as follows:

$$A_\alpha = \{\omega \in \Omega | A(\omega) \geq \alpha\}$$

The empty set $\emptyset$ can be viewed as a fuzzy set to which the membership degree of each element in Ω is equal to 0.
The union of two fuzzy sets A and B defined over Ω is a fuzzy set C of Ω written as $C = A \cup B$ and its is defined by:

$$C(\omega) = \max(A(\omega), B(\omega)), \quad \forall\, \omega \in \Omega.$$

Let C represent now the intersection of A and B, denoted as $C = A \cap B$. C is a fuzzy set of Ω defined by:

$$C(\omega) = \min(A(\omega), B(\omega)), \quad \forall\, \omega \in \Omega.$$

3.2.1.2 Properties of fuzzy sets

The fuzzy set operations defined above have many properties in common with their crisp counterparts, such as commutativity, associativity, distributivity, transitivity, idempotency, De Morgan's laws, etc. More precisely:

- Commutativity:

$$A \cup B = B \cup A,$$

$$A \cap B = B \cap A.$$

- Associativity:

$$A \cup (B \cup C) = (A \cup B) \cup C,$$

$$A \cap (B \cap C) = (A \cap B) \cap C.$$

- Distributivity:

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C),$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C).$$

- Idempotency:

$$A \cup A = A,$$

$$A \cap A = A.$$

- Identity:

$$A \cup \emptyset = A,$$

$$A \cap \Omega = A.$$

- Absorption:

$$A \cup (A \cap B) = A,$$

$$A \cap (A \cup B) = A.$$

- Involution:

$$\overline{\overline{A}} = A.$$

- De Morgan's laws:

$$\overline{A \cup B} = \overline{A} \cap \overline{B},$$

$$\overline{A \cap B} = \overline{A} \cup \overline{B}.$$

3.2.2 Possibility theory

Possibility theory was first introduced by Zadeh based on fuzzy set theory [124]. Before presenting an overview about this theory, we have to note that it includes two variants: quantitative (numerical) and qualitative. These two variants mainly differ by the conditioning operation [35]. This section is devoted to a review about quantitative possibility theory. In the following, it will simply be referred to as possibility theory.

3.2.2.1 Possibility distribution

Let v denote a possibilistic variable taking one and only one value in Ω. For example, Ω is a finite set of classes, and v is the class label of an instance to classify. A *possibility distribution* π on Ω is a mapping from Ω to the unit interval $[0, 1]$:

$$\pi : \Omega \longrightarrow [0, 1].$$

It represents a piece of knowledge about v. The identity $\pi(\omega) = 0$ means that $\omega \in \Omega$ is an impossible value of v and it is totally excluded, while $\pi(\omega) = 1$ just means that $v = \omega$ is normal and unsurprising and is one of the most possible values of v. A possibility distribution on Ω can be regarded as the membership function of a fuzzy subset of Ω [124]. π is said to be normalized if $\pi(\omega) = 1$ for at least one element ω in Ω, in which case Ω is considered to be exhaustive [33].

Complete knowledge about v is represented by a possibility distribution π such that $\pi(\omega_0) = 1$ for some element $\omega_0 \in \Omega$, and $\pi(\omega) = 0$ for $\omega \neq \omega_0$. The situation of complete ignorance about the true value of v is represented by a possibility distribution π such that $\pi(\omega) = 1$ for each element ω in Ω. Given two possibility distribution π and π' representing two pieces of knowledge about v, we say that π' is *more informative* or *more specific* than π if, for each element $\omega \in \Omega$, $\pi'(\omega) \leq \pi(\omega)$ [118]. The set of possible values of v according to π' is then more restricted than the set of possible values of v according to π [29]. The possibility distribution π such that $\pi(\omega) = 1$ for all $\omega \in \Omega$, is the greatest element of this partial ordering relation.

3.2.2.2 Possibility and Necessity measures

Two measures on Ω can be derived from π. They are called *possibility* and *necessity* measures and they are denoted by Π and N, respectively. Formally, the possibility measure is the mapping from the power set of Ω to the interval [0,1], defined by:

$$\Pi : 2^{\Omega} \longrightarrow [0, 1],$$

$$\Pi(A) = \sup_{\omega \in A} \pi(\omega), \ \forall A \subseteq \Omega$$

The necessity measure is defined as follows:

$$N : 2^{\Omega} \longrightarrow [0, 1],$$

$$N(A) = 1 - \Pi(\overline{A}) = \inf_{\omega \notin A} (1 - \pi(\omega)),\ \forall A \subseteq \Omega$$

The number $\Pi(A)$ represents the degree of possibility of the proposition (or event) "$v \in A \subseteq \Omega$", while $N(A)$ represents the degree of necessity (certainty) of that proposition. In other words, $\Pi(A)$ measures to what extent at least one element in A is possible, and $N(A)$ measures to what extent no element not belonging to A is possible.

Possibility measures satisfy the "maxitivity" property, i.e, the possibility degree of a disjunction of events is the maximum of the possibility degrees of these events:

$$\Pi\left(\bigcup_{i=1}^{n} A_i\right) = \max(\Pi(A_i), i = 1, \ldots, n).$$

Dually, the necessity degree of a conjunction of events is the minimum of the necessity degrees of the events:

$$N\left(\bigcap_{i=1}^{n} A_i\right) = \min(N(A_i), i = 1, \ldots, n).$$

3.2.2.3 Combination of possibility distributions

Given different possibility distributions $\pi_i, (i = 1, \ldots, n)$ representing knowledge about the value of v, there exist different combination rules to aggregate these distributions [5]. The basic combination rules are the conjunctive and disjunction ones. Let π_{conj} and π_{disj} be the possibility distributions obtained by combining the π_i's, $i = 1, \ldots, n$, conjunctively and disjunctively, respectively. We have:

$$\pi_{conj}(\omega) = \min(\pi_i(\omega), i = 1, \ldots, n), \quad \forall\ \omega \in \Omega,$$

and,

$$\pi_{disj}(\omega) = \max(\pi_i(\omega), i = 1, \ldots, n), \quad \forall\ \omega \in \Omega.$$

In general, the conjunctive rule is used when all pieces of knowledge are considered to be reliable, while the disjunctive rule corresponds to a weaker reliability hypothesis. If for all $\omega \in \Omega$, $\pi_{conj}(\omega)$ is much smaller than 1, we can infer that at least one of the combined pieces of knowledge is likely to be wrong, and the disjunctive rule may be more adequate to that case.

3.2.2.4 Comparison with Probability Theory

In Probability Theory, probability measures are *self-dual* in the sense that $\Pr(A) = 1-\Pr(\overline{A})$. In contrast, in possibility theory, necessity measures are the dual of possibility measures, as we have $N(A) = 1-\Pi(\overline{A})$. Given a possibility measure Π and a probability measure Pr, Pr is said to be covered by Π if:

$$\Pr(A) \leq \Pi(A), \ \forall A \subseteq \Omega.$$

This relation means that what is possible *may not* be probable, while what is impossible *is also* improbable [36].

Notions of conditioning and independence have been proposed for possibility measures. By analogy with probability theory, we may define:

$$\Pi(A|B) = \frac{\Pi(A \cap B)}{\Pi(B)},$$

and

$$N(A|B) = 1 - \Pi(\overline{A}|B),$$

for each $A \subseteq \Omega$ and each $B \subseteq \Omega$ such that $\Pi(B) \neq 0$ [3].

3.2.2.5 Certainty-qualified knowledge

Usually, pieces of knowledge about the true value of v are expressed in a way that some trust qualification is attached. Certainty-qualified pieces of knowledge about the true value of v are of the form "v *is* A is $\alpha-$certain", where $\alpha \in [0,1]$ represents the degree of certainty of the proposition "v *is* A". Note that *is* is a relation to represent knowledge about possibilistic variables [34]. If A is a crisp subset of Ω, such piece of certainty-qualified knowledge means that it is certain at least at the degree α that the value of v is in A, or, equivalently, that any value outside A is at most possible to the complementary degree $1-\alpha$ [33]. A possibility distribution π on Ω can be induced from such piece of knowledge verifying the following constraints:

$$\pi(\omega) \leq \max(A(\omega), 1-\alpha), \text{ for all } \omega \in \Omega,$$

and we have,

$$N(A) \geq \alpha.$$

The principle of minimum specificity results in attributing possibility 1 to values in A, and $1-\alpha$ to values not in A, and to assign α as certainty degree to the proposition "v *is* A". When α increases from 0 to 1, our knowledge evolves from complete ignorance about v to complete certainty in "v *is* A".

In the general case of fuzzy subsets of Ω, the piece of knowledge "v *is* A is $\alpha-$certain" leads to "v *is* B" such that, the membership function of B can be defined as [34]:

$$B(\omega) = \max(A(\omega), 1-\alpha), \text{ for all } \omega \in \Omega.$$

3.3 Veristic variables

In [119], Yager develops a theory for the expression within the language of approximate reasoning of statements involving veristic variables, i.e., variables taking as values fuzzy subsets of the universe of discourse.

3.3.1 Veristic statements

Let Ω denote a universe of discourse, and V a variable taking zero, one or several values in Ω, i.e, V takes a single value in the set I^{Ω} of fuzzy subsets of Ω. Such a variable is said to be veristic. Let $V_0 \in I^{\Omega}$ denote the unknown true value of V. Giving a fuzzy set $A \in I^{\Omega}$, the following statements can be made to associate variable V with A [119]:

1. V *isv* A, meaning that $A \subseteq V_0$;
2. V *isv*(n) A, meaning that $V_0 \subseteq \overline{A}$;
3. V *isv*(c) A, meaning that $V_0 = A$;
4. V *isv*(c, n) A, meaning that $V_0 = \overline{A}$.

In the above expressions, the relation *isv* has two parameters: c for closed and n for negative. The following example gives an illustration of these notations.

Example 1 For a multi-label classification problem, assume that instances are songs and classes are emotions generated by these songs, as in the emotion dataset used in the experiments reported in Chapter 5. Upon hearing a song, more than one emotion can be generated at the same time. Let V be a variable that corresponds to the emotions evoked by a given song. Let A be the set containing the emotions "sad" and "quiet".

- V *isv* A, means that the song evokes sadness and quietness but it can also generate other emotions such as anger, calm, surprise, etc. This statement represents an open positive (or affirmative) information;
- V $isv(n)$ A, means that the song evokes neither sadness nor quietness. We have no idea about the remaining emotions. This is an open negative information;
- V $isv(c)$ A, means that the song *only* evokes sadness and quietness, no more emotions being generated by this song. This is a closed (or exclusive) positive information;
- V $isv(c,n)$ A, means that the song *only does not* evoke sadness and quietness. This is a closed negative information.

As remarked by Yager, any piece of knowledge about a veristic variable V of the form V $isv(.)$ A, can be interpreted by specifying a crisp or fuzzy set W of fuzzy subsets of Ω, such that W contains the possible values of V consistent with that knowledge. For each $B \in I^{\Omega}$, $W(B)$ is the degree of membership of B in W. The most simple representation is to consider W as a crisp subset of I^{Ω}, and thus, for the statement V *isv* A, $W(B) = 1$ if $B \supseteq A$, and $W(B) = 0$ if $B \not\supseteq A$ [119]. Hereafter, we give the *crisp* definition of W for the four types of veristic statement:

1. V *isv* $A \rightarrow W = \{B \in I^{\Omega} | B \supseteq A\}$;
2. V $isv(n)$ $A \rightarrow W = \{B \in I^{\Omega} | B \subseteq \overline{A}\}$;
3. V $isv(c)$ $A \rightarrow W = \{A\}$;
4. V $isv(c,n)$ $A \rightarrow W = \{\overline{A}\}$.

3.3.2 Verity and Rebuff distributions

From the veristic statement V $isv(.)$ A, two functions from Ω to $[0,1]$ associated to the corresponding set W can be induced, allowing us to provide information about the different elements of the frame of discourse Ω. These functions, called the *verity* and $rebuff$ distributions, are defined as follow:

$$\mathrm{Ver}(\omega) = \min_{B \in I^{\Omega}} \max(B(\omega), \overline{W}(B)),$$

and,

$$\text{Rebuff}(\omega) = 1 - \max_{B \in I^{\Omega}} \min(B(\omega), W(B)),$$

for each $\omega \in \Omega$.

In the special case in which W is a crisp subset, the definitions of verity and rebuff distributions are reduced to:

$$\text{Ver}(\omega) = \min_{B \in W} B(\omega),$$

and,

$$\text{Rebuff}(\omega) = 1 - \max_{B \in W} B(\omega) = \min_{B \in W} \overline{B}(\omega),$$

for each $\omega \in \Omega$.

In the following, we will only consider the case where W is a crisp set of fuzzy subsets of Ω. $\text{Ver}(\omega)$ is then the minimal membership degree of ω in any subset in W, while $\text{Rebuff}(\omega)$ is the minimal membership degree of ω in the complement of any subset in W. $\text{Ver}(\omega)$ can thus be viewed as the minimal support for ω being one of the values taken by V, while, $\text{Rebuff}(\omega)$ can be interpreted as the minimal support for ω *not* being one of the values taken by V.

In [120], a possibility distribution Poss has been also introduced. For each element ω in Ω, $\text{Poss}(\omega)$ represents the *maximal* support for ω being one of the values taken by V. $\text{Poss}(\omega)$ is the complement of $\text{Rebuff}(\omega)$:

$$\text{Poss}(\omega) = 1 - \text{Rebuff}(\omega) = \max_{B \in W} B(\omega), \quad \forall\, \omega \in \Omega.$$

We can remark that $\text{Ver}(\omega)$ represents a lower bound on the truth of the proposition "ω is one of the solutions of V", while $\text{Poss}(\omega)$ represents an upper bound on the truth of that proposition. We can deduce that $\text{Ver}(\omega) \leq \text{Poss}(\omega)$, from which it follows that

$$\text{Ver}(\omega) + \text{Rebuff}(\omega) \leq 1, \quad \forall\, \omega \in \Omega. \tag{3.1}$$

The state of total ignorance about a veristic variable V is represented by verity and rebuff distributions such as: $\text{Ver}(\omega) = 0$ and $\text{Rebuff}(\omega) = 0$, for all $\omega \in \Omega$. Complete knowledge about V can be represented as follows: $\max(\text{Ver}(\omega), \text{Rebuff}(\omega)) = 1$ and $\min(\text{Ver}(\omega), \text{Rebuff}(\omega)) = 0$, for each element ω.

The verity and rebuff distributions have the following expressions for the different veristic statement types:

1. $V\ isv\ A \Rightarrow \text{Ver}(\omega) = A(\omega)$ and $\text{Rebuff}(\omega) = 0,\ \forall \omega \in \Omega$;
2. $V\ isv(n)\ A \Rightarrow \text{Ver}(\omega) = 0$ and $\text{Rebuff}(\omega) = A(\omega),\ \forall \omega \in \Omega$;
3. $V\ isv(c)\ A \Rightarrow \text{Ver}(\omega) = A(\omega)$ and $\text{Rebuff}(\omega) = 1 - A(\omega),\ \forall \omega \in \Omega$;
4. $V\ isv(c, n)\ A \Rightarrow \text{Ver}(\omega) = 1 - A(\omega)$ and $\text{Rebuff}(\omega) = A(\omega),\ \forall \omega \in \Omega$.

For instance, giving the open veristic statement $V\ isv\ A$, the set W of fuzzy subsets of Ω representing the possible solutions of V is $W = \{B \in I^{\Omega} | B \supseteq A\}$. For any subset B in W, we have $B(\omega) \geq A(\omega)$, $\forall\ \omega \in \Omega$. Thus, for each element $\omega \in \Omega$, the minimal degree of membership of ω to a fuzzy subset in W is $A(\omega)$, and the maximal degree of membership is 1 because Ω belongs to W as we have $\Omega \supseteq A$ and $\Omega(\omega) = 1$. Therefore, the verity measure $\text{Ver}(\omega)$ of each element $\omega \in \Omega$ is $\min\limits_{B \in W} B(\omega) = A(\omega)$, and the rebuff measure $\text{Rebuff}(\omega)$ of ω is $1 - \max\limits_{B \in W} B(\omega) = 0$.

In comparison with possibility theory, Poss defines a possibility measure, while Ver and Rebuff define necessity measures. The difference is that Ver and Rebuff are not defined on subsets of Ω but, individually, on the elements of Ω.

In the following, we will pay special attention to open veristic statements, which will be more relevant for our purpose.

3.3.3 Combination of veristic information

Given two pieces of knowledge about a veristic variable V, the *conjunctive* combination of the corresponding veristic statements is defined as follows:

$$V\ isv\ A_1\ \textbf{and}\ V\ isv\ A_2\ \equiv\ V\ isv\ A_1 \cup A_2,$$

$$V\ isv(n)\ A_1\ \textbf{and}\ V\ isv(n)\ A_2\ \equiv\ V\ isv(n)\ A_1 \cup A_2.$$

The *disjunctive* combination of veristic statements is defined by:

$$V\ isv\ A_1\ \textbf{or}\ V\ isv\ A_2\ \equiv\ V\ isv\ A_1 \cap A_2,$$

$$V\ isv(n)\ A_1\ \textbf{or}\ V\ isv(n)\ A_2\ \equiv\ V\ isv(n)\ A_1 \cap A_2.$$

We notice the unexpected association of union and intersection with the conjunctive and disjunctive combination, respectively. Usually, as for example the Dempster's rule of combination in belief function theory [93], the conjunctive combination of different

pieces of knowledge is associated with intersection and not with union as it is the case here.

Example 2 As in Example 1, let V be a variable representing the emotions evoked by a given song. A first expert tells us that the song evokes emotions *amazed* and *happy*. For a second expert, the evoked emotions are *amazed* and *angry*. If we trust both experts, the conjunctive combination of the two pieces of knowledge leads to the conclusion that the emotions evoked by the song are *amazed*, *happy* and *angry*. In contrast, if only one of the two experts is reliable, the disjunctive combination is recommended. Thus, the emotion that corresponds to the song is *amazed*.

In this chapter, we are interested in the combination of veristic knowledge modeled by verity and rebuff distributions. Let Ver_1 and Rebuff_1 be the verity and rebuff distributions that correspond to the first source of information about the veristic variable V, and let Ver_2 and Rebuff_2 be the corresponding distributions of the second source of information about V. Let Ver and Rebuff denote the resulted distributions after combination.

The disjunctive combination of the informations given by the two sources of knowledge is defined as follows:

$$\mathrm{Ver}(\omega) = (\mathrm{Ver}_1 \textbf{ or } \mathrm{Ver}_2)(\omega) = \min(\mathrm{Ver}_1(\omega), \mathrm{Ver}_2(\omega)),$$

and,

$$\mathrm{Rebuff}(\omega) = (\mathrm{Rebuff}_1 \textbf{ or } \mathrm{Rebuff}_2)(\omega) = \min(\mathrm{Rebuff}_1(\omega), \mathrm{Rebuff}_2(\omega)),$$

for each $\omega \in \Omega$.

The conjunctive combination of the informations given by the two sources of knowledge is defined as follows:

$$\mathrm{Ver}(\omega) = (\mathrm{Ver}_1 \textbf{ and } \mathrm{Ver}_2)(\omega) = \max(\mathrm{Ver}_1(\omega), \mathrm{Ver}_2(\omega)),$$

and,

$$\mathrm{Rebuff}(\omega) = (\mathrm{Rebuff}_1 \textbf{ and } \mathrm{Rebuff}_2)(\omega) = \max(\mathrm{Rebuff}_1(\omega), \mathrm{Rebuff}_2(\omega)),$$

for each $\omega \in \Omega$.

We have to be careful when using the conjunctive combination, because we risk to violate the assumption (3.1) if there is a *conflict* between the two pieces of information to combine. In fact, if for an element $\omega \in \Omega$, $\text{Ver}(\omega) + \text{Rebuff}(\omega) > 1$, which means that the two sources of knowledge are conflicting and cannot be combined. This risk does not exist when combining the information disjunctively.

Note that the conjunctive (**and**) and disjunctive (**or**) combination rules are commutative and associative.

We now propose another combination rule that will be denoted by **and/or** and that is defined as:

$$\begin{aligned} \text{Ver}(\omega) = & \quad (\text{Ver}_1 \textbf{ and/or } \text{Ver}_2)(\omega) \\ = & \begin{cases} (\text{Ver}_1 \textbf{ or } \text{Ver}_2)(\omega) & \text{if } \text{Ver}(\omega) + \text{Rebuff}(\omega) > 1 \\ (\text{Ver}_1 \textbf{ and } \text{Ver}_2)(\omega) & \text{otherwise,} \end{cases} \end{aligned}$$

and,

$$\begin{aligned} \text{Rebuff}(\omega) = & \quad (\text{Rebuff}_1 \textbf{ and/or } \text{Rebuff}_2)(\omega) \\ = & \begin{cases} (\text{Rebuff}_1 \textbf{ or } \text{Rebuff}_2)(\omega) & \text{if } \text{Ver}(\omega) + \text{Rebuff}(\omega) > 1 \\ (\text{Rebuff}_1 \textbf{ and } \text{Rebuff}_2)(\omega) & \text{otherwise,} \end{cases} \end{aligned}$$

for each $\omega \in \Omega$.

The proposed *hybrid* rule of combination allows us to combine different informations conjunctively while avoiding the risk of having conflict. This rule is commutative but not associative. The hybrid rule is inspired from the Dubois and Prade combination rule [32] in the framework of belief function theory [95].

3.3.4 Discounting

Let $\alpha \in [0, 1]$ be the degree of certainty or reliability associated with the given statement about a veristic variable V. The equality $\alpha = 1$ implies that the given information is fully reliable, while for $\alpha = 0$, the knowledge will be discarded. We suppose that the given knowledge is represented by verity and rebuff distributions: Ver and Rebuff. By taking the parameter α into account, the discounted distributions denoted by Ver^α and Rebuff^α are defined as:

$$\text{Ver}^\alpha(\omega) = \min(\text{Ver}(\omega), \alpha),$$

and,

$$\text{Rebuff}^{\alpha}(\omega) = \min(\text{Rebuff}(\omega), \alpha),$$

for each $\omega \in \Omega$.

The discounted possibility distribution Poss^{α} is:

$$\text{Poss}^{\alpha}(\omega) = \max(1 - \text{Rebuff}(\omega), 1 - \alpha).$$

These equations mean that if the source of information is reliable at degree α, the corresponding verity and rebuff measures are at most equal to α, and the possibility measures are at least equal to $1 - \alpha$. The discounting introduced here is very close to the discounting of certainty-qualified knowledge for possibilistic variables explained in Subsection 3.2.2.5.

Example 3 Let V be the label set of a giving song. An expert tell us that this song certainly evoke *happiness* and certainly does not evoke *sadness*. In the framework of veristic variables, this knowledge is represented as follow: $\text{Ver}(\textit{happiness}) = 1$, $\text{Rebuff}(\textit{sadness}) = 1$, and the verity and rebuff values of the remaining emotions are equal to 0. If we have a 80% ($\alpha = 0.8$) confidence in the opinion of the expert, the new verity and rebuff distributions after discounting are: $\text{Ver}^{\alpha}(\textit{happiness}) = 0.8$, $\text{Rebuff}^{\alpha}(\textit{sadness}) = 0.8$, and the verity and rebuff values of the remaining emotions remain equal to 0.

3.4 Multi-label learning based on veristic variable framework

In this section, we propose a k-NN rule for multi-label learning using the theory of veristic variables presented in Section 3.3. k-NN rules, discussed in Section 2.3, are widely used in classification problems due to their simplicity and their competitiveness with other sophisticated learning methods. The proposed algorithm is called VERkNN for Veristic k-Nearest Neighbor. Two major issues are related to our proposed method. The first one concerns the influence of the nearest neighbors on the classification of an unseen instance $\mathbf{x}$. Each neighbor represents a piece of knowledge about the classification result of $\mathbf{x}$ where instead of giving equal importance to all neighbors as in the voting k-NN rule, a weight or a degree of certainty is assigned to each neighbor according to the distance to $\mathbf{x}$ [60][133]. The second issue concerns the class membership of training

data. In the framework of veristic variables, the knowledge about the labeling of each instance can be represented by a verity distribution representing positive information, and a rebuff distribution representing negative information.
The issue of the labeling of training instances will be discussed in Section 3.4.1. The VERkNN algorithm will be then introduced in Section 3.4.2.

3.4.1 Labeling of training data

Let $\mathbb{X}$ denote the domain of instances and $\mathcal{Y} = \{\omega_1, \omega_2, \ldots, \omega_{\mathrm{Q}}\}$ the finite set of labels. Usually, the available datasets to train multi-label classifiers are constructed in such a way that each instance $\mathbf{x}_i$ is *perfectly* labeled, i.e., $\mathbf{x}_i$ is associated with a crisp subset Y_i of $\mathcal{Y}$. However, such situations are not always possible and feasible at a reasonable cost, and may be especially questioned when training data are labeled by one or several experts. In practice, due to lack of confidence and absence of ground truth, an expert may be undecided about the labeling of a given instance. He may then express positive information about the labels that should be attributed to the given instance, and negative information about the labels that should not be attributed to that instance. Thus, the expert will be unable to assign unambiguously a crisp label set to each instance (see Example 4). The veristic variable framework seems adequate to represent and manipulate such information.

Example 4 Let $\mathcal{Y} = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5\}$ be the set of classes, and let $\mathbf{x}$ be an instance labeled by an expert. The expert tells us that $\mathbf{x}$ certainly belongs to class ω_1 and certainly does not belong to class ω_2. He is sure at 60% that $\mathbf{x}$ should also be assigned to class ω_3, and with a certainty equal to 75% that $\mathbf{x}$ should not be assigned to class ω_4. The expert is totally undecided about the membership of $\mathbf{x}$ to class ω_5. The labeling of $\mathbf{x}$ can be represented by the following verity and rebuff vectors: $\mathrm{Ver} = (1, 0, 0.6, 0, 0)$ and $\mathrm{Rebuff} = (0, 1, 0, 0.75, 0)$.

Let $\mathcal{D} = \{(\mathbf{x}_1, Y_1), \ldots, (\mathbf{x}_n, Y_n)\}$ be a perfectly labeled training dataset, where $\mathbf{x}_i \in \mathbb{X}$ and $Y_i \subseteq \mathcal{Y}$. We present hereafter two approaches, a *direct* approach and a *fuzzy* one, which allow us to label training instances by verity and rebuff measures instead of crisp sets of labels.

3.4.1.1 Direct approach

For each precisely labeled training object $(\mathbf{x}_i, Y_i)$, the corresponding veristic object $(\mathbf{x}_i, \mathrm{Ver}_i, \mathrm{Rebuff}_i)$ can be derived as follows:

$$\begin{aligned} \mathrm{Ver}_i(\omega) &= \begin{cases} 1 & \text{if } \omega \in Y_i \\ 0 & \text{otherwise} \end{cases} \\ &= Y_i(\omega) \end{aligned}$$

and,

$$\begin{aligned} \mathrm{Rebuff}_i(\omega) &= \begin{cases} 1 & \text{if } \omega \notin Y_i \\ 0 & \text{otherwise} \end{cases} \\ &= 1 - Y_i(\omega) \end{aligned}$$

for each $\omega \in \mathcal{Y}$.

Note that, these definitions extend directly to the case where Y_i is a fuzzy subset of $\mathcal{Y}$.

3.4.1.2 Fuzzy approach

For each training instance $\mathbf{x}_i$, verity and rebuff distributions can also be determined by taking into account the neighborhood of this instance. Let k' denote the number of neighbors to be considered in order to determine Ver_i and Rebuff_i. Let $\mathcal{N}_{\mathbf{x}_i}^{k'}$ denote the k' nearest neighbors of $\mathbf{x}_i$ in the training dataset $\mathcal{D}$. For a class $\omega \in \mathcal{Y}$, let $p^i = (p_0^i, p_1^i)$ be the probability distribution such as p_1^i (respectively, p_0^i) denote the proportion of instances in $\mathcal{N}_{\mathbf{x}_i}^{k'}$ which belong (respectively, do not belong) to class ω. We have:

$$p_1^i = \frac{|\{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}^{k'} |\ \omega \in Y_j\}|}{k'},$$

$$p_0^i = \frac{|\{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}^{k'} |\ \omega \notin Y_j\}|}{k'},$$

and,

$$p_1^i + p_0^i = 1.$$

In accordance with the possibilistic interpretation of a veristic variable, the probability distribution $p^i = (p_0^i, p_1^i)$ can be transformed into a possibility distribution $\pi^i = (\pi_0^i, \pi_1^i)$ using a *probability-possibility transformation*. π_0^i is the possibility of the proposition "$\mathbf{x}_i$

does not belong to ω", and π_1^i the possibility of "$\mathbf{x}_i$ belongs to ω". Several probability-possibility transformations exist in the literature [61][30]. In this work, the transformation introduced in [36] will be used. In the following, we recall the principle of this transformation that will be referred here to as *Prob/Poss*-transformation.

The *Prob/Poss*-transformation Let $p = (p_1, p_2, \ldots, p_n)$ be a probability distribution such as $p_1 \geq p_2 \geq \ldots \geq p_n$, and let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ the corresponding possibility distribution. Using the *Prob/Poss*-transformation, π is the solution that verifies the following constraints:

- $\text{Pr}(H) \leq \Pi(H)$, for each hypothesis or proposition H, where Pr (respectively, Π) is the probability (respectively, possibility) measure derived from p (respectively, π);
- p and π are order-equivalent, i.e., if $p_q \geq p_r$, then $\pi_q \geq \pi_r$;
- π is maximally specific (or informative), i.e. for any other solution π', we have $\pi_q \leq \pi'_q,\ \forall\ q \in \{1, \ldots, n\}$.

The possibility distribution π satisfying these requirements is unique and it is derived from p as follows [36]:

$$\pi_1 = 1,$$

and,

$$\pi_q = \begin{cases} \sum_{r=q}^{n}\ p_r & \text{if } p_q < p_{q-1}, \\ \pi_{q-1} & \text{otherwise.} \end{cases}$$

Based on the *Prob/Poss*-transformation, π_0^i and π_1^i are computed as follow:

$$\begin{cases} \text{If } p_1^i > p_0^i, & \pi_1^i = 1 \ \text{ and } \pi_0^i = p_0^i; \\ \text{If } p_1^i < p_0^i, & \pi_1^i = p_1^i \text{ and } \pi_0^i = 1; \\ \text{If } p_1^i = p_0^i, & \pi_1^i = 1 \ \text{ and } \pi_0^i = 1. \end{cases}$$

Thus, for the class $\omega \in \mathcal{Y}$, the verity and rebuff values for the training instance $\mathbf{x}_i$ are defined as:

$$\begin{cases} \text{Ver}_i(\omega) = 1 - \pi_0^i, \\ \text{Rebuff}_i(\omega) = 1 - \pi_1^i. \end{cases}$$

We can explain these relations by the fact that, in the context of possibility theory, the verity and rebuff distributions define *necessity* measures. As shown in Section 3.2.2, the necessity $N(H)$ of a proposition H is related to the possibility $\Pi(H)$ of this proposition by the following equation:

$$N(H) = 1 - \Pi(\overline{H}).$$

For the class $\omega \in \mathcal{Y}$, $\text{Ver}_i(\omega)$ represents the necessity of the proposition H_1 "$\mathbf{x}_i$ belongs to ω", and $\text{Rebuff}_i(\omega)$ represents the necessity of the proposition H_0 "$\mathbf{x}_i$ does not belong to ω". Thus, $\text{Ver}_i(\omega)$ is the complement of the possibility π_0^i of H_0, and $\text{Rebuff}_i(\omega)$ the complement of the possibility π_1^i of H_1.

3.4.2 Proposed method: VERkNN

The VERkNN method builds a multi-label classifier $\mathcal{H} : \mathbb{X} \longrightarrow 2^{\mathcal{Y}}$ and a scoring-function $f : \mathbb{X} \times \mathcal{Y} \to \mathbb{R}$ from a training dataset $\mathcal{D}$ that is assumed to be of the form $\mathcal{D} = \{(\mathbf{x}_1, \text{Ver}_1, \text{Rebuff}_1), \ldots, (\mathbf{x}_n, \text{Ver}_n, \text{Rebuff}_n)\}$, where $\mathbf{x}_i \in \mathcal{X}$, and the corresponding labeling is represented by the two distributions Ver_i and Rebuff_i that define mappings from the set $\mathcal{Y}$ to the interval $[0, 1]$.

Let $\mathbf{x}$ be an unseen instance for which we search to estimate the set of labels. The classification of $\mathbf{x}$ is performed by exploiting the information of its k nearest neighbors in $\mathcal{D}$. The proposed method performs as follows:

1. Search for the k nearest neighbors of $\mathbf{x}$ in $\mathcal{D}$, represented by $\mathcal{N}_{\mathbf{x}}^k$, based on a certain distance function $d(.,.)$, usually the Euclidean one.

2. Each element $(\mathbf{x}_i, \text{Ver}_i, \text{Rebuff}_i)$ in $\mathcal{N}_{\mathbf{x}}^k$ represents a piece of knowledge about the labeling of $\mathbf{x}$. The influence of $\mathbf{x}_i$ on the classification of $\mathbf{x}$ depends on the distance between $\mathbf{x}$ and $\mathbf{x}_i$. If $\mathbf{x}_i$ is *close* to $\mathbf{x}$ according to the distance function $d(.,.)$, then one will be inclined to believe that both instances have the same labeling. Let α_i represents the degree of certainty associated with the knowledge given by $(\mathbf{x}_i, \text{Ver}_i, \text{Rebuff}_i)$ on the labeling of $\mathbf{x}$. If $d(\mathbf{x}, \mathbf{x}_i)$ decreases and tends to 0, α_i increases and tends to 1. In our method, as in [20], the value of α_i is determined using the following equation:

$$\alpha_i = \alpha_0 \exp(-\gamma d(\mathbf{x}, \mathbf{x}_i)), \tag{3.2}$$

with $0 < \alpha_0 < 1$ and $\gamma > 0$. Parameter α_0 is fixed at a value close to 1 such as $\alpha_0 = 0.95$, whereas γ should depend on the scaling of distances and can be either fixed heuristically or optimized [20].

3. The verity and rebuff distributions of each element $(\mathbf{x}_i, \mathrm{Ver}_i, \mathrm{Rebuff}_i)$ in $\mathcal{N}_{\mathbf{x}}^k$ are updated using the corresponding parameter α_i. That leads to the discounted piece of knowledge $(\mathbf{x}_i, \mathrm{Ver}_i^{\alpha_i}, \mathrm{Rebuff}_i^{\alpha_i})$.

4. Combine the verity distributions and the rebuff distributions of the k nearest neighbors of $\mathbf{x}$. Let Ver and Rebuff represent the aggregated verity and rebuff distributions, respectively. We have:

$$\mathrm{Ver} = \mathrm{Ver}_1^{\alpha_1} * \ldots * \mathrm{Ver}_k^{\alpha_k},$$

and

$$\mathrm{Rebuff} = \mathrm{Rebuff}_1^{\alpha_1} * \ldots * \mathrm{Rebuff}_k^{\alpha_k},$$

where $*$ denote the combination operator: **and**, **or**, or **and**/**or**. It seems preferable to use the hybrid or the disjunctive rules of combination in order to avoid conflict. Note that, when using the hybrid rule for combination, we have to fix an *order* to combine the informations about the labeling of $\mathbf{x}$ coming from its different neighbors, as the hybrid rule is not associative. The combination can be done by going from the nearest neighbor to the furthest one, by going in the reverse order, or by using a random order.

5. The output of VERkNN is determined as follow:

$$\mathcal{H}(\mathbf{x}) = \{\omega \in \mathcal{Y} |\ \mathrm{Ver}(\omega) > \mathrm{Rebuff}(\omega)\},$$

and,

$$f(\mathbf{x}, \omega) = \mathrm{Ver}(\omega).$$

3.5 Conclusion

In this chapter, we have presented a k-nearest neighbor rule for multi-label learning using the framework of veristic variables. This framework allows us to represent different pieces of knowledge about a veristic variable by different types of statements

and distributions, and combine them conjunctively or disjunctively, in order to make decision about the values taken by this variable. By considering the class label of each instance as a veristic variable, we have used this theory to build a multi-label classification method called VERkNN. Each unseen instance is classified on the basis of its k nearest neighbors. The labeling of each training instance is represented by a verity distribution representing positive information, and a rebuff distribution representing negative information. The verity and rebuff distributions are discounted depending on the distance to the instance to classify, and are then combined in order to determine the classes to assigned to the unseen instance. This method is proposed to solve multi-label classification problems where training datasets are labeled by one or several experts in the absence of ground truth, and the opinions of experts about the class label of training data are represented by verity and rebuff vectors. It will be evaluated with the other method proposed in this thesis in Chapter 4.

Chapter 4

Set-valued evidence formalism and application to multi-label learning

Summary

In this chapter, we propose an evidence formalism for representing and handling partial knowledge about set-valued variables, based on the Dempster-shafer theory of belief functions. Set-valued variables are variables that can take more than one value at the same time, such as the class label of a multi-labeled instance. Given a set-valued variable defined over a universe Ω, the straightforward approach is to consider it as a single-valued variable taking one and only one value in 2^Ω. To represent uncertainty about this variable, we have to define mass functions on the frame 2^{2^Ω}, which is usually not feasible because of the double-exponential complexity involved. Our formalism consists in defining a restricted family of subsets of 2^Ω which is closed under intersection and has a lattice structure. Using recent results about belief functions on lattices, we show that most notions from Dempster-Shafer theory can be transposed to that particular lattice, making it possible to express rich knowledge about set-valued variables with only limited additional complexity as compared to the single-valued case. Based on the proposed formalism, we introduce an evidential multi-label classification method, where each instance is classified on the basis of its k nearest neighbors from a given training set. This method is proposed to solve multi-label classification problems where training data are imprecisely labeled.

Résumé

Dans ce chapitre, nous proposons un formalisme de croyance pour la représentation et la manipulation de connaissances partielles concernant des variables multi-valuées à l'aide de la théorie des fonctions de croyance de Dempster-Shafer. Les variables multi-valuées sont des variables qui peuvent avoir plusieurs valeurs en même temps, comme par exemple, l'étiquette d'un individu dans un problème d'apprentissage multi-label. Étant donnée une variable multi-valuée définie sur un univers Ω, l'approche directe et intuitive consiste à considérer cette variable comme étant une variable mono-valuée prenant une et une seule valeur dans l'ensemble 2^{Ω}. Une connaissance partielle à propos de cette variable sera représentée par une fonction de masse définie sur l'ensemble $2^{2^{\Omega}}$. Cette approche directe nous amène à travailler dans un espace de très grande dimension, ce qui n'est pas toujours faisable vue la double complexité exponentielle impliquée. L'idée de base du formalisme de croyance que nous proposons est de ne pas considérer l'ensemble $2^{2^{\Omega}}$ tout entier, mais juste un sous-ensemble clos par intersection et ayant une structure de treillis. En utilisant des résultats récents concernant la définition des fonctions de croyance sur des treillis, nous montrons que la plupart des notions de base de la théorie de Dempster-shafer peuvent être transposées à ce sous-ensemble particulier, permettant d'exprimer suffisamment de connaissances partielles sur des variables multi-valuées avec seulement une légère augmentation de complexité par rapport au cas de manipulation de variables mono-valuées. Nous montrons aussi l'application de ce formalisme conjointement avec l'approche des k plus proches voisins pour l'apprentissage multi-label. Cette méthode est destinée pour la classification des individus avec des étiquettes imprécises.

4.1 Introduction

In this chapter, we consider the problem of representing partial knowledge about a set-valued variable V with domain Ω using the Dempster-Shafer theory of belief functions [93][99]. This theory is one of the principle techniques for representing and handling uncertainty in decision making.

A straightforward approach to the above problem is, of course, to consider a set-valued variable V on Ω as a single-valued variable on the power set $\Theta = 2^{\Omega}$. However, this approach often implies working in a space of very high cardinality. If, as done in this chapter, we assume Ω to be finite, then the size of Θ is $2^{|\Omega|}$. If we want to express imprecise information about V, we will have to manipulate subsets of Θ. As there are $2^{2^{|\Omega|}}$ of these subsets, this approach rapidly becomes intractable as the size of Ω increases.

Our approach will be based on a simple representation of a class $\mathcal{C}(\Omega)$ of subsets of $\Theta = 2^{\Omega}$ which, endowed with set inclusion, has a lattice structure. Using recent results about belief functions on lattices [49], we will be able to generalize most concepts of Dempster-Shafer theory (including the canonical decompositions and the cautious rule [21]) in this setting. This formalism will be shown to allow the expression of a wide range of knowledge about set-valued variables, with only a moderate increase of complexity (from $2^{|\Omega|}$ to $3^{|\Omega|}$) as compared to the usual single-valued case.

Originally, the motivation behind this work was to build a multi-label classifier using the evidence theory in order to handle uncertainties and ambiguities when classifying unseen instances. The class label of a multi-labeled instance is an example of set-valued variables. An evidential k-NN rule, called EMLkNN, will be presented in this chapter using the proposed formalism.

This chapter is organized as follows. Background notions on belief functions in the classical setting and in general lattices will first be recalled in Sections 4.2 and 4.3, respectively. Our approach will then be introduced in Section 4.4, and some relationships with previous work will be outlined in Section 4.5. An application to multi-label classification will be presented in Section 4.6, and Section 4.7 will conclude the chapter.

4.2 Belief Functions

The basic concepts of the Dempster-Shafer theory of belief functions, as introduced in [93], will first be summarized in Subsection 4.2.1. The canonical decomposition and the cautious rule will then be recalled in Subsection 4.2.2.

4.2.1 Basic definitions

Let Ω be a finite set. A *mass function* on Ω is a function $m : 2^\Omega \to [0, 1]$ such that

$$\sum_{A \subseteq \Omega} m(A) = 1.$$

The subsets A of Ω such that $m(A) > 0$ are called the *focal elements* of m. The set of focal elements of m will be denoted $\mathcal{F}(m)$. m is said to be *normal* if $\emptyset$ is not a focal set, and *dogmatic* if Ω is not a focal set.

A mass function m is often used to model an agent's beliefs about a variable V taking a single but ill-known value ω_0 in Ω [99]. The quantity $m(A)$ is then interpreted as the measure of the belief that is committed *exactly* to the hypothesis $\omega_0 \in A$. Full certainty corresponds to the case where $m(\{\omega_q\}) = 1$ for some $\omega_q \in \Omega$, while total ignorance is modelled by the *vacuous* mass function verifying $m(\Omega) = 1$.

To each mass function m can be associated an *implicability function* b and a *belief function* bel defined as follows:

$$b(A) = \sum_{B \subseteq A} m(B) \tag{4.1}$$

$$bel(A) = \sum_{B \subseteq A, B \not\subseteq \overline{A}} m(B) = b(A) - m(\emptyset). \tag{4.2}$$

These two functions are equal when m is normal. However, they need to be distinguished when considering non normal mass functions. Function bel has easier interpretation, as $bel(A)$ corresponds to a degree of belief in the proposition "The true value ω_0 of V belongs to A". However, function b has simpler mathematical properties. For instance, m can be recovered from b as

$$m(A) = \sum_{B \subseteq A} (-1)^{|A \setminus B|} b(B). \tag{4.3}$$

Function m is said to be the *Möbius transform* of b. For every function f from 2^Ω to $[0, 1]$ such that $f(\Omega) = 1$, the following conditions are known to be equivalent [93]:

1. The Möbius transform m of f is positive and verifies $\sum_{A\subseteq\Omega} m(A) = 1$.

2. f is totally monotone, i.e., for any $q \geq 2$ and for any family $A_1, \ldots, A_q$ in 2^Ω,

$$f\left(\bigcup_{i=1}^{q} A_i\right) \geq \sum_{\emptyset\neq I\subseteq\{1,\ldots,q\}} (-1)^{|I|+1} f\left(\bigcap_{i\in I} A_i\right).$$

Hence, b (and bel) are totally mononotone.

Other functions related to m are the *plausibility function*, defined as

$$pl(A) = \sum_{B\cap A\neq\emptyset} m(B) \tag{4.4}$$

$$= 1 - b(\overline{A}) \tag{4.5}$$

and the *commonality function* (or co-Möbius transform of b) defined as

$$q(A) = \sum_{B\supseteq A} m(B). \tag{4.6}$$

m can be recovered from q using the following relation:

$$m(A) = \sum_{B\supseteq A} (-1)^{|B\setminus A|} q(B). \tag{4.7}$$

Functions m, bel, b, pl and q are thus in one-to-one correspondence and can be regarded as different facets of the same information.

Two special cases of interest have to be mentioned:

1) If all focal elements of a mass function m are *singletons*, m is equivalent to a probability distribution on Ω, and corresponds to probabilistic uncertainty.

2) If the focal elements of m are *nested*, m is then said to be *consonant* and it is equivalent to a possibility distribution π on Ω, defined as: $\pi(\omega) = pl(\{\omega\})$ for all $\omega \in \Omega$. In fact, in the case of consonant mass functions, we have:

$$pl(A \cup B) = \max(pl(A), pl(B)), \quad \forall A, B \subseteq \Omega.$$

The plausibility function pl derived from m is thus a possibility measure Π corresponding to π. Conversely, to each possibility distribution corresponds a unique consonant mass function [93].

Let us now assume that we receive two mass functions m_1 and m_2 from two distinct sources of information assumed to be reliable. Then m_1 and m_2 can be combined using the *conjunctive sum* (or unnormalized Dempster's rule of combination) defined as follows:

$$(m_1 \textcircled{\cap} m_2)(A) = \sum_{B \cap C = A} m_1(B) m_2(C). \tag{4.8}$$

This rule is commutative, associative, and admits the vacuous mass function as neutral element. It is conjunctive as the product of $m_1(B)$ and $m_2(C)$ is transferred to the intersection of B and C. The quantity $(m_1 \textcircled{\cap} m_2)(\emptyset)$ is referred to as the *degree of conflict* between m_1 and m_2.

Let $q_{1\textcircled{\cap}2}$ denote the commonality function corresponding to $m_1 \textcircled{\cap} m_2$. It can be computed from q_1 and q_2, the commonality functions associated to m_1 and m_2, as follows:

$$q_{1\textcircled{\cap}2}(A) = q_1(A) \cdot q_2(A), \quad \forall A \subseteq \Omega. \tag{4.9}$$

The normalized Dempster's rule $\oplus$ [93] is defined as the conjunctive sum followed by a normalization step:

$$(m_1 \oplus m_2)(A) = \begin{cases} 0 & \text{if } A = \emptyset \\ \frac{(m_1 \textcircled{\cap} m_2)(A)}{1 - (m_1 \textcircled{\cap} m_2)(\emptyset)} & \text{otherwise.} \end{cases} \tag{4.10}$$

It is clear that $m_1 \oplus m_2$ is defined as long as $(m_1 \textcircled{\cap} m_2)(\emptyset) < 1$.

Alternatives to the conjunctive sum can be constructed by replacing $\cap$ by any binary set operation in (4.8). For instance, the choice of the union operator results in the *disjunctive sum* [97]:

$$(m_1 \textcircled{\cup} m_2)(A) = \sum_{B \cup C = A} m_1(B) m_2(C). \tag{4.11}$$

It can be shown that

$$b_{1\textcircled{\cup}2}(A) = b_1(A) \cdot b_2(A), \quad \forall A \subseteq \Omega, \tag{4.12}$$

which is the counterpart of (4.9). Dubois and Prade [28] have also proposed a "hybrid" rule intermediate between the conjunctive and disjunctive sums, in which the product $m_1(B)m_2(C)$ is assigned to $B \cap C$ whenever $B \cap C \neq \emptyset$, and to $B \cup C$ otherwise. This rule is not associative, but it usually provides a good summary of partially conflicting items of evidence.

In [99], Smets proposed a two-level model in which items of evidence are quantified by mass functions and combined at the *credal* level, while decisions are made at the *pignistic* level (from the Latin *pignus* meaning a bet). Once a decision has to be made, a mass function m is thus transformed into a *pignistic probability distribution* p. The pignistic transformation consists in normalizing m (assuming that $m(\emptyset) < 1$), and then distributing each normalized mass $m(A)/(1-m(\emptyset))$ equally between the atoms $\omega_k \in A$:

$$p(\omega_q) = \sum_{\{A \subseteq \Omega, \omega_q \in A\}} \frac{m(A)}{(1-m(\emptyset))|A|}, \quad \forall \omega_q \in \Omega. \tag{4.13}$$

Other authors have suggested the so-called plausibility transformation for transforming a mass function into a probability distribution, by normalizing the plausibilities of singletons [14]. In a decision making context, this approach results in selecting the most plausible single hypothesis.

4.2.2 Canonical Decompositions and Idempotent Rules

According to Shafer [93], a mass function is said to be *simple* if it has the following form

$$\begin{aligned} m(A) &= 1 - w_0 \\ m(\Omega) &= w_0, \end{aligned}$$

for some $A \subset \Omega$ and some $w_0 \in [0, 1]$. Let us denote such a mass function as A^{w_0}. The vacuous mass function may thus be noted A^1 for any $A \subset \Omega$. It is clear that

$$A^{w_0} \textcircled{\cap} A^{w_0'} = A^{w_0 w_0'}.$$

A mass function may be called *separable* if it can be obtained as the result of the conjunctive sum of simple mass functions. It can then be written:

$$m = \textcircled{\cap}_{A \subset \Omega} A^{w(A)}, \tag{4.14}$$

with $w(A) \in [0, 1]$ for all $A \subset \Omega$.

Smets [98] showed that any non dogmatic mass function m can be uniquely expressed using (4.14), with weights $w(A)$ now in $(0, +\infty)$. This is referred to as the *conjunctive canonical decomposition* of a mass function. Note that, when $w(A) > 1$, $A^{w(A)}$ is no longer a mass function, but the conjunctive sum can be extended to such "generalized mass functions" in an obvious way.

Function w is called the *conjunctive weight function* associated to m [21]. It is a new equivalent representation of a non dogmatic mass function, which may be computed directly from q as follows:

$$w(A) = \prod_{B \supseteq A} q(B)^{(-1)^{|B \setminus A|+1}}, \quad \forall A \subset \Omega, \tag{4.15}$$

or, taking logarithms,

$$\ln w(A) = -\sum_{B \supseteq A} (-1)^{|B \setminus A|} \ln q(B), \quad \forall A \subset \Omega. \tag{4.16}$$

In [98] and [21], $w(A)$ was defined for all strict subsets A of Ω. However, function w can be extended to 2^Ω by using (4.15) for $A = \Omega$. We then have:

$$w(\Omega) = \frac{1}{q(\Omega)} = \frac{1}{m(\Omega)} = \left(\prod_{A \subset \Omega} w(A) \right)^{-1}$$

and

$$\prod_{A \subseteq \Omega} w(A) = 1. \tag{4.17}$$

With this convention, (4.16) can be extended to all $A \subseteq \Omega$. We notice that (4.16) then has exactly the same form as (4.7), i.e., the formula for computing $\ln w$ from $-\ln q$ is the same as the one for computing m from q. Conversely, $\ln q$ can thus be computed from $-\ln w$ using a formula similar to (4.6):

$$\ln q(A) = -\sum_{B \supseteq A} \ln w(B), \quad \forall A \subseteq \Omega.$$

We note that function w has a simple property with respect to the conjunctive sum. Let w_1 and w_2 be two weight functions, and let $w_{1 \textcircled{\cap} 2}$ denote the result of their $\textcircled{\cap}$-combination. Then the following relation holds:

$$w_{1 \textcircled{\cap} 2}(A) = w_1(A) w_2(A), \quad \forall A \subseteq \Omega. \tag{4.18}$$

In [21], Denœux introduced the *cautious rule*, noted $\textcircled{\wedge}$, which is obtained by replacing the product by the minimum in (4.18), for all $A \subset \Omega$:

$$w_{1 \textcircled{\wedge} 2}(A) = \min\left(w_1(A), w_2(A)\right). \tag{4.19}$$

The value of $w_{1\textcircled{\wedge}2}(\Omega)$ can then be determined to satisfy the normalization condition (4.17). This rule is obviously commutative, associative and idempotent. As shown in [21], it is suitable for combining conjunctively non independent items of evidence. As the conjunctive sum, the cautious rule has a normalized version defined by

$$(m_1 \textcircled{\wedge}^* m_2)(A) = \begin{cases} 0 & \text{if } A = \emptyset \\ \frac{(m_1 \textcircled{\wedge} m_2)(A)}{1-(m_1 \textcircled{\wedge} m_2)(\emptyset)} & \text{otherwise.} \end{cases} \tag{4.20}$$

As shown in [21], the conjunctive canonical decomposition also has a disjunctive counterpart. Any mass function m such that $m(\emptyset) > 0$ can be decomposed disjunctively as follows:

$$m = \textcircled{\cup}_{A \supset \emptyset} A_{v(A)}, \tag{4.21}$$

where $A_{v(A)}$ is a generalized mass function assigning a mass $v(A) > 0$ (possibly greater than 1) to $\emptyset$, and $1 - v(A)$ to A, for all $A \subseteq \Omega$, $A \neq \emptyset$. This defines a new function v, called the *disjunctive weight function*, which can be computed from b as follows:

$$v(A) = \prod_{B \subseteq A} b(B)^{(-1)^{|A \setminus B|+1}}, \quad \forall A \subseteq \Omega, A \neq \emptyset, \tag{4.22}$$

or

$$\ln v(A) = -\sum_{B \subseteq A} (-1)^{|A \setminus B|} \ln b(B), \quad \forall A \subseteq \Omega, A \neq \emptyset. \tag{4.23}$$

As before, the above equations can be extended to $A = \emptyset$, which leads to

$$v(\emptyset) = \frac{1}{b(\emptyset)} = \frac{1}{m(\emptyset)} = \left(\prod_{A \neq \emptyset} v(A) \right)^{-1}$$

and

$$\prod_{A \subseteq \Omega} v(A) = 1. \tag{4.24}$$

The disjunctive rule (4.11) has a simple expression as a function of disjunctive weights:

$$v_{1\textcircled{\cup}2}(A) = v_1(A) v_2(A), \quad \forall A \subseteq \Omega. \tag{4.25}$$

By replacing the product by the minimum in the above equation, we can define a new rule, denoted $\textcircled{\vee}$ and called the *bold rule* in [21]:

$$v_{1\textcircled{\vee}2}(A) = \min\left(v_1(A), v_2(A)\right), \quad A \subseteq \Omega, A \neq \emptyset, \tag{4.26}$$

and $v_{1\textcircled{\vee}2}(\emptyset) = \left(\prod_{A \neq \emptyset} v_{1\textcircled{\vee}2}(A) \right)^{-1}$. This rule is obviously commutative, associative and idempotent; it is suitable for combining disjunctively non independent items of evidence.

4.3 Extension to General Lattices

As shown by Grabisch [49], the theory of belief function can be extended from the Boolean lattice $(2^\Omega, \subseteq)$ to any lattice, not necessarily Boolean. We will first recall some basic definitions about lattices in Subsection 4.3.1. Grabisch's results used in this work will then be summarized in Subsection 4.3.2.

4.3.1 Lattices

A review of lattice theory can be found in [78]. The following presentation follows [49].

Let L be a finite set and $\leq$ a partial ordering (i.e., a reflexive, antisymmetric and transitive relation) on L. The structure $(L, \leq)$ is called a *poset*. We say that $(L, \leq)$ is a *lattice* if, for every $x, y \in L$, there is a unique greatest lower bound (denoted $x \wedge y$) and a unique least upper bound (denoted $x \vee y$). Operations $\wedge$ and $\vee$ are called the *meet* and *join* operations, respectively. For finite lattices, the greatest element (denote $\top$) and the least element (denoted $\bot$) always exist. We say that x *covers* y if $x > y$ and there is no z such that $x > z > y$. An element x of L is an *atom* if it covers only one element and this element is $\bot$. It is a *co-atom* if it is covered by a single element and this element is $\top$.

Two lattices L and L' are *isomorphic* if there exists a bijective mapping f from L to L' such that $x \leq y \Leftrightarrow f(x) \leq f(y)$. For any poset $(L, \leq)$, we can define its dual $(L, \geq)$ by inverting the order relation. A lattice is *autodual* if it is isomorphic to its dual.

A lattice is *distributive* if $(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z)$ holds for all $x, y, z \in L$. For any $x \in L$, we say that x has a complement in L if there exists $x' \in L$ such that $x \wedge x' = \bot$ and $x \vee x' = \top$. L is said to be *complemented* if any element has a complement. Boolean lattices are distributive and complemented lattices. Every Boolean lattice is isomorphic to $(2^\Omega, \subseteq)$ for some set Ω. For the lattice $(2^\Omega, \subseteq)$, we have $\wedge = \cap$, $\vee = \cup$, $\bot = \emptyset$ and $\top = \Omega$.

A *closure system* $\mathcal{C}$ on a set Θ is a family of subsets of Θ satisfying the following properties:

1. $\Theta \in \mathcal{C}$.

2. $C_1, C_2 \in \mathcal{C} \Rightarrow C_1 \cap C_2 \in \mathcal{C}$.

As shown in [78], any closure system $(\mathcal{C}, \subseteq)$ is a lattice with the following meet and join operations

$$C_1 \wedge C_2 = C_1 \cap C_2 \tag{4.27}$$

$$C_1 \vee C_2 = \bigcap \{C \in \mathcal{C} | C_1 \cup C_2 \subseteq C\}. \tag{4.28}$$

4.3.2 Belief Functions on Lattices

Let $(L, \leq)$ be a finite poset having a least element, and let f be a function from L to $\mathbb{R}$. The *Möbius transform* of f is the function $m : L \to \mathbb{R}$ defined as the unique solution of the equation:

$$f(x) = \sum_{y \leq x} m(y), \quad \forall x \in L. \tag{4.29}$$

Function m can be expressed as:

$$m(x) = \sum_{y \leq x} \mu(y, x) f(y), \tag{4.30}$$

where $\mu(x, y) : L^2 \to \mathbb{R}$ is the *Möbius function* defined inductively by:

$$\mu(x, y) = \begin{cases} 1 & \text{if } x = y, \\ -\sum_{x \leq t < y} \mu(x, t) & \text{if } x < y, \\ 0, & \text{otherwise.} \end{cases} \tag{4.31}$$

The *co-Möbius transform* of f is defined as:

$$q(x) = \sum_{y \geq x} m(y), \tag{4.32}$$

and m can be recovered from q as:

$$m(x) = \sum_{y \geq x} \mu(x, y) q(y). \tag{4.33}$$

Let us now assume that $(L, \leq)$ is a lattice. Following Grabisch [49], a function $b : L \to [0, 1]$ will be called an implicability function on L if $b(\top) = 1$, and its Möbius transform is non negative. The corresponding belief function *bel* can then be defined as:

$$bel(x) = b(x) - m(\bot), \quad \forall x \in L.$$

Note that Grabisch [49] considered only normal belief functions, in which case $b = bel$. As shown in [49], any implicability function on $(L, \leq)$ is totally monotone, i.e., for any $k \geq 2$ and for any family $x_1, \ldots, x_k$ in L,

$$b\left(\bigvee_{i=1}^{k} x_i\right) \geq \sum_{\emptyset \neq I \subseteq \{1,\ldots,k\}} (-1)^{|I|+1} b\left(\bigwedge_{i \in I} x_i\right).$$

Note, however, that the converse does not hold in general: a totally monotone function may not have a non negative Möbius transform.

As shown in [49], most results of Dempster-Shafer theory can be transposed in the general setting of lattices. For instance, the conjunctive sum (4.8) becomes:

$$(m_1 \textcircled{\cap} m_2)(x) = \sum_{y \wedge z = x} m_1(y) m_2(z), \quad \forall x \in L, \tag{4.34}$$

and the following relation between commonality functions still holds:

$$q_{1\textcircled{\cap}2}(x) = q_1(x) \cdot q_2(x), \quad \forall x \in L. \tag{4.35}$$

The normalized Dempster's rule $\oplus$ can still be defined, as in the classical case, by dividing each number $(m_1 \textcircled{\cap} m_2)(x)$ with $x \neq \bot$ by $1 - (m_1 \textcircled{\cap} m_2)(\bot)$, provided that $(m_1 \textcircled{\cap} m_2)(\bot) < 1$.

Using a similar line of reasoning as that followed in [49], we can also extend the disjunctive rule (4.11) as:

$$(m_1 \textcircled{\cup} m_2)(x) = \sum_{y \vee z = x} m_1(y) m_2(z), \quad \forall x \in L, \tag{4.36}$$

and (4.12) becomes:

$$b_{1\textcircled{\cup}2}(x) = b_1(x) \cdot b_2(x), \quad \forall x \in L. \tag{4.37}$$

Grabisch [49] also extended the conjunctive canonical decomposition of belief functions in the general lattice setting. He showed that any mass function m on L such that $m(\top) > 0$ can be decomposed as

$$m = \textcircled{\cap}_{x < \top} x^{w(x)}, \tag{4.38}$$

where $x^{w(x)}$ is a simple mass function assigning $1 - w(x)$ to V and $w(x)$ to $\top$, with $w(x) \in (0, +\infty)$. Clearly, (4.38) generalizes (4.14). As in the classical case, function $w : L \setminus \{\top\} \to (0, +\infty)$ can be computed from q using the following equation:

$$w(x) = \prod_{y \geq x} q(y)^{-\mu(x,y)}, \quad \forall x \in L, x \neq \top, \tag{4.39}$$

which generalizes (4.15). Obviously, we still have

$$w_{1\textcircled{\cap}2}(x) = w_1(x)w_2(x), \quad \forall x \in L, x \neq \top. \tag{4.40}$$

The existence of the w function also allows us to define the cautious rule in the general lattice setting as

$$w_{1\textcircled{\wedge}2}(x) = \min\left(w_1(x), w_2(x)\right), \quad \forall x \in L, x \neq \top. \tag{4.41}$$

The normalized cautious rule $\textcircled{\wedge}^*$ is defined as in the classical case, by dividing each $w_{1\textcircled{\wedge}2}(x)$ for $x \neq \bot$ by $1 - w_{1\textcircled{\wedge}2}(\bot)$, provided that $w_{1\textcircled{\wedge}2}(\bot) < 1$.

Although Grabisch did not consider the disjunctive canonical decomposition, it can also be extended in the general lattice setting. The proof parallels that given in [49] for the conjunctive case. We will only state the main result here. Let $x_{v(x)}$ be a mass function on L assigning $1 - v(x)$ to V and $v(x)$ to $\bot$, with $v(x) \in (0, +\infty)$. Any mass function m on L such that $m(\bot) > 0$ can be decomposed as

$$m = \textcircled{\cup}_{x>\bot} x_{v(x)}. \tag{4.42}$$

The function $v : L \setminus \{\bot\} \to (0, +\infty)$ can be computed from b using the following equation:

$$v(x) = \prod_{y \leq x} b(y)^{-\mu(y,x)}, \quad \forall x \in L, x \neq \bot, \tag{4.43}$$

which generalizes (4.22). We still have

$$v_{1\textcircled{\cup}2}(x) = v_1(x)v_2(x), \quad \forall x \in L, x \neq \bot, \tag{4.44}$$

and the existence of the v function allows us to define the bold rule as

$$v_{1\textcircled{\vee}2}(x) = \min\left(v_1(x), v_2(x)\right), \quad \forall x \in L, x \neq \bot. \tag{4.45}$$

The extension of other notions from classical Dempster-Shafer theory may require additional assumptions on $(L, \leq)$. For instance, the definition of the plausibility function *pl* as the dual of *b* using (4.5) can only be extended to autodual lattices [49]. The definition of *pl* from (4.4) remains possible in the other cases, but the relationship between *pl* and *b* (or *bel*) is lost. Also, probability measures cannot be defined on arbitrary lattices. Consequently, the pignistic probability (4.13) can only be extended in restricted settings.

Remark 1 Although our approach relies essentially on Grabisch's work, we may note the existence of another line of research that aims at extending results of Probability Theory to some classes of residuated lattices, which are more general than Boolean algebra. In particular, there have been many developments about probability measures on MV-algebra (also called *states*), see, e.g., [12][63][64][80] as well as in Gödel algebras [1]. In addition, a recent work on defining belief functions on MV-algebras is introduced in [65].

4.4 Belief Functions on Set-valued Variables

In this section, the main concepts of Dempster-Shafer theory recalled in Section 4.2 will be extended to the case where we want to describe the uncertainty regarding a set-valued variable V on a finite domain Ω. The key to this extension will be the definition of a closure system $\mathcal{C}(\Omega)$ of $\Theta = 2^\Omega$, i.e., a set of subsets of Θ that is closed under intersection. Each element of $\mathcal{C}(\Omega)$ will be shown to have a simple description as a pair of disjoint subsets of Ω. Belief functions and associated notions will then be defined on the lattice $(\mathcal{C}(\Omega), \subseteq)$, resulting in a simple framework for uncertain reasoning about set-valued variables.

4.4.1 The Lattice $(\mathcal{C}(\Omega), \subseteq)$

Let V denote a set-valued variable on a finite domain Ω, and let $A_0 \subseteq \Omega$ be the unknown true value of V. We want to describe partial knowledge about that value in the belief function framework.

As explained in the introduction, the formalism recalled in Section 4.2 could be applied without modification to this case, by defining a mass function m^Θ on Θ. However, such a brute force approach would require the storage of up to $2^{|\Theta|} = 2^{2^{|\Omega|}}$ numbers for each mass function. Basic operations such as the conjunctive or disjunctive sums would have double-exponential complexity, making the approach inapplicable except for sets Ω with very small cardinality.

As an alternative, we propose to define mass functions and associated functions on a subset of 2^Θ that forms a lattice when equipped with the inclusion relation. The intuitive idea underlying our approach is the fact that, when expressing knowledge about a set-valued variable V, it is often convenient to specify sets of values that are

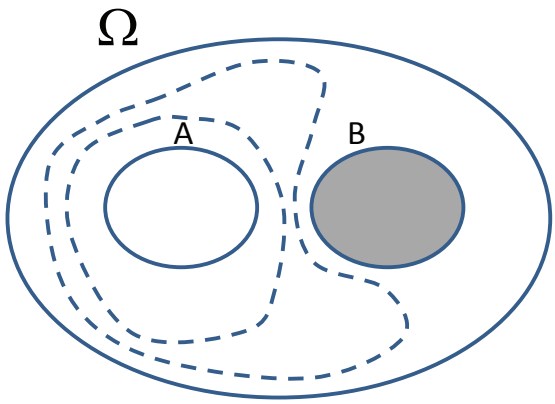

Figure 4.1: Two subsets of Ω (broken lines) containing A and not intersecting B. The set of all such subsets is denoted by $\varphi(A, B)$.

certainly taken by V, and sets of values that are *certainly not* taken by V. This can be illustrated by the following example.

Example 5 Let V denote the languages spoken by John, defined on the (very large) set Ω of existing languages. If we know for sure that John can speak English and French (because he was brought up in the US and he stayed in France for a long time), and that he can speak neither Japanese nor Chinese (because he never traveled to Asia), then all subsets of Ω containing $A = \{English, French\}$ and not intersecting $B = \{Japanese, Chinese\}$ are possible values of V.

As shown by this example, some families of subsets of Ω or, equivalently, some subsets of $\Theta = 2^\Omega$ can be conveniently described by two subsets A and B of Ω such that $A \cap B = \emptyset$ (Figure 4.1).

More generally, let $\mathcal{Q}(\Omega) = \{(A, B) \in 2^\Omega \times 2^\Omega | A \cap B = \emptyset_\Omega\}$ be the set of ordered pairs of disjoint subsets of Ω, where $\emptyset_\Omega$ denotes the empty set of Ω. For any $(A, B) \in \mathcal{Q}(\Omega)$, let $\varphi(A, B)$ denote the following subset of $\Theta = 2^\Omega$:

$$\varphi(A, B) = \{C \subseteq \Omega | C \supseteq A, C \cap B = \emptyset_\Omega\}. \tag{4.46}$$

$\varphi(A, B)$ is thus the subset of Θ composed of all subsets of Ω including A and non intersecting B. Equivalently, it is the set of all subsets of Ω that include A and are included in $\overline{B}$:

$$\varphi(A, B) = \{C \subseteq \Omega | A \subseteq C \subseteq \overline{B}\}. \tag{4.47}$$

It is thus the interval $[A, \overline{B}]$ in the lattice $(\Omega, \subseteq)$.

Let $\mathcal{C}(\Omega)$ denote the set of all subsets of Θ of the form $\varphi(A, B)$, completed by the empty set of Θ, noted $\emptyset_\Theta$:

$$\mathcal{C}(\Omega) = \{\varphi(A, B) | A \subseteq \Omega, B \subseteq \Omega, A \cap B = \emptyset_\Omega\} \cup \{\emptyset_\Theta\}.$$

$\mathcal{C}(\Omega)$ is thus a subset of 2^Θ. For a reason that will become evident later, we will also use $\varphi(\Omega, \Omega)$ as an alternative notation for $\emptyset_\Theta$. Function φ is thus a bijective mapping from $\mathcal{Q}^*(\Omega) = \mathcal{Q}(\Omega) \cup \{(\Omega, \Omega)\}$ to $\mathcal{C}(\Omega)$. The following proposition states that $\mathcal{C}(\Omega)$ is a closure system and, consequently, has a lattice structure.

Proposition 1 $\mathcal{C}(\Omega)$ is a closure system of Θ, and

$$\varphi(A, B) \cap \varphi(A', B') = \begin{cases} \varphi(A \cup A', B \cup B') & \text{if } (A \cup A') \cap (B \cup B') = \emptyset_\Omega \\ \emptyset_\Theta & \text{otherwise,} \end{cases}$$

for all (A, B) and (A', B') in $\mathcal{Q}^*(\Omega)$.

Proof: It is obvious that $\Theta = \varphi(\emptyset_\Omega, \emptyset_\Omega) \in \mathcal{C}(\Omega)$. Now,

$$\begin{aligned} \varphi(A, B) \cap \varphi(A', B') &= \{C \subseteq \Omega | C \supseteq A, C \supseteq A', C \cap B = \emptyset_\Omega, C \cap B' = \emptyset_\Omega\} \\ &= \{C \subseteq \Omega | C \supseteq (A \cup A'), C \cap (B \cup B') = \emptyset_\Omega\}. \end{aligned}$$

If $(A \cup A') \cap (B \cup B') = \emptyset_\Omega$ then $\varphi(A, B) \cap \varphi(A', B')$ is thus equal to $\varphi(A \cup A', B \cup B')$. Otherwise, no subset C of Ω can include $A \cup A'$ and have an empty intersection with $B \cup B'$; consequently, $\varphi(A, B) \cap \varphi(A', B') = \emptyset_\Theta$. □

As recalled in Section 4.3.1, any closure system endowed with the inclusion relation has a lattice structure with $\wedge = \cap$ and $\vee$ defined by (4.28). Here, the inclusion relation has the following simple expression using the $\varphi(A, B)$ representation:

$$\varphi(A, B) \subseteq \varphi(A', B') \Leftrightarrow A \supseteq A' \text{ and } B \supseteq B'. \tag{4.48}$$

The least element is $\bot = \varphi(\Omega, \Omega) = \emptyset_\Theta$. We note that (4.48) remains valid when $A = B = \Omega$, which explains the interest of the notation $\varphi(\Omega, \Omega) = \emptyset_\Theta$. The greatest element is $\top = \varphi(\emptyset_\Omega, \emptyset_\Omega) = \Theta$. The atoms are of the form $\varphi(A, \overline{A})$ for $A \subseteq \Omega$, and the co-atoms are of the form $\varphi(\{\omega\}, \emptyset_\Omega)$ or $\varphi(\emptyset_\Omega, \{\omega\})$ for $\omega \in \Omega$. We can see that the number of atoms is not equal to the number of co-atoms, which shows that $(\mathcal{C}, \subseteq)$ is not autodual. This lattice is also not complemented; consequently, it is not Boolean.

As a consequence of (4.48), it is easy to see that the meet operation $\vee$ is the following operation, hereafter denoted $\sqcup$:

$$\varphi(A, B) \sqcup \varphi(A', B') = \varphi(A \cap A', B \cap B').$$

It must be noted that $\sqcup$ is not identical to set union. The following proposition states the relation between these two operators.

Proposition 2 For all (A, B) and (A', B') in $\mathcal{Q}^*(\Omega)$,

$$\varphi(A, B) \cup \varphi(A', B') \subseteq \varphi(A, B) \sqcup \varphi(A', B').$$

Proof: For every C in $\varphi(A, B) \cup \varphi(A', B')$, we have

$$C \supseteq A \text{ and } C \supseteq A' \Rightarrow C \supseteq A \cap A' \tag{4.49}$$

and

$$C \cap B = \emptyset_\Omega \text{ and } C \cap B' = \emptyset_\Omega \Rightarrow C \cap (B \cap B') = \emptyset_\Omega, \tag{4.50}$$

hence $C \in \varphi(A \cap A', B \cap B')$. □

One can notice that the implications in (4.49) and (4.50) are strict. Consequently, $\varphi(A, B) \cup \varphi(A', B')$ is usually a strict subset of $\varphi(A, B) \sqcup \varphi(A', B')$. As the lattices $(\mathcal{C}(\Omega), \subseteq)$ and $(2^\Theta, \subseteq)$ do not have the same join operator, $(\mathcal{C}(\Omega), \subseteq)$ is not a sublattice of $(2^\Theta, \subseteq)$, although it is a subposet.

As noticed in [50], any ordered pair (A, B) of disjoint subsets of $\Omega = \{\omega_1, \ldots, \omega_Q\}$ can be represented by a vector $(y_1, \ldots, y_Q) \in \{-1, 0, 1\}^Q$, with

$$y_i = \begin{cases} 1 & \text{if } \omega_i \in A, \\ -1 & \text{if } \omega_i \in B, \\ 0 & \text{otherwise.} \end{cases}$$

Consequently, any $\varphi(A, B) \in \mathcal{C}(\Omega)$ such that $(A, B) \neq (\Omega, \Omega)$ can be represented in the same way. For $\varphi(\Omega, \Omega) = \emptyset_\Theta$, a special representation can be adopted, e.g., $(*, \ldots, *)$. This encoding makes it possible to implement the $\cap$ and $\sqcup$ operations in a simple way using generalized truth tables. It also makes it clear that the cardinality of $\mathcal{C}(\Omega)$ is equal to $3^{|\Omega|} + 1$.

4.4.2 Belief Functions on $\mathcal{C}(\Omega)$

The general theory recalled in Section 4.3.2 can be applied directly to the lattice $(\mathcal{C}(\Omega), \subseteq)$.

Let $m : \mathcal{C}(\Omega) \to [0, 1]$ be a mass function on $\mathcal{C}(\Omega)$. The notation $m(\varphi(A, B))$ will be simplified to $m(A, B)$. For this reason, m will be called a *two-place mass function*. We assume that

$$\sum_{(A,B)\in\mathcal{Q}^*(\Omega)} m(A, B) = 1.$$

The implicability, belief and commonality functions can be computed from m using the following formula:

$$b(A, B) = \sum_{\varphi(C,D)\subseteq\varphi(A,B)} m(C, D) = \sum_{C\supseteq A, D\supseteq B} m(C, D), \tag{4.51}$$

$$bel(A, B) = b(A, B) - m(\Omega, \Omega), \tag{4.52}$$

$$q(A, B) = \sum_{\varphi(C,D)\supseteq\varphi(A,B)} m(C, D) = \sum_{C\subseteq A, D\subseteq B} m(C, D), \tag{4.53}$$

where all pairs (A, B) and (C, D) are understood to belong to $\mathcal{Q}^*(\Omega)$ (the same convention will be adopted throughout this chapter). The conjunctive sum operation in $\mathcal{C}(\Omega)$ is defined as follows:

$$(m_1 \textcircled{\cap} m_2)(A, B) = \sum_{\varphi(C,D)\cap\varphi(E,F)=\varphi(A,B)} m_1(C, D)m_2(E, F) \tag{4.54}$$

$$= \begin{cases} \sum_{C\cup E=A, D\cup F=B} m_1(C, D)m_2(E, F) & \text{if } A \cap B = \emptyset_\Omega, \\ \sum_{(C\cup E)\cap(D\cup F)\neq\emptyset_\Omega} m_1(C, D)m_2(E, F) & \text{if } A = B = \Omega. \end{cases} \tag{4.55}$$

It can be computed using the commonality functions as:

$$q_{1\textcircled{\cap}2}(A, B) = q_1(A, B) \cdot q_2(A, B), \quad \forall (A, B) \in \mathcal{Q}^*(\Omega). \tag{4.56}$$

Similarly, the disjunctive sum can be computed as:

$$(m_1 \textcircled{\cup} m_2)(A, B) = \sum_{\varphi(C,D)\sqcup\varphi(E,F)=\varphi(A,B)} m_1(C, D)m_2(E, F) \tag{4.57}$$

$$= \sum_{C\cap E=A, D\cap F=B} m_1(C, D)m_2(E, F), \tag{4.58}$$

or using implicability functions:

$$b_{1\textcircled{\scriptsize \cup}2}(A,B) = b_1(A,B) \cdot b_2(A,B), \quad \forall (A,B) \in \mathcal{Q}^*(\Omega).$$

It is also possible to define a rule expressing a consensus among items of evidence, somehow in the same spirit as the Dubois-Prade rule recalled in Section 4.2.1. Assume that we learn from two sources that the value of V is in $\varphi(C,D)$ and in $\varphi(E,F)$, but that $\varphi(C,D) \cap \varphi(E,F) = \emptyset_\Theta$, i.e., $(C \cup E) \cap (D \cup F) \neq \emptyset_\Omega$, so that the two pieces of information are in conflict. It may still be safe to keep $(C \cup E) \setminus (D \cup F)$ as positive information, and $(D \cup F) \setminus (C \cup E)$ as negative information. Denoting by $\sqcap$ the following operation on $\mathcal{C}(\Omega)$:

$$\varphi(C,D) \sqcap \varphi(E,F) = \varphi\left((C \cup E) \setminus (D \cup F), (D \cup F) \setminus (C \cup E)\right),$$

we may define a new combination rule as

$$(m_1 \boxdot m_2)(A,B) = \sum_{\varphi(C,D) \sqcap \varphi(E,F) = \varphi(A,B)} m_1(C,D) m_2(E,F). \tag{4.59}$$

This rule will be referred to as the *consensus rule*. We note that operations $\sqcap$ and $\boxdot$ are not associative. However, they are quasi-associative, as it is possible to define a n-ary version of $\sqcap$ as:

$$\varphi(C_1,D_1) \sqcap \ldots \sqcap \varphi(C_n,D_n) = \varphi\left(\bigcup_{i=1}^n C_i \setminus \bigcup_{i=1}^n D_i, \bigcup_{i=1}^n D_i \setminus \bigcup_{i=1}^n C_i\right).$$

To compute functions m, w and v from q or b using (4.30), (4.33), (4.39) and (4.43), we need the expression of the Möbius function μ. It is given in the following proposition.

Proposition 3 The Möbius function on $(\mathcal{C}(\Omega), \subseteq)$ is given, for any (A,B) and (A',B') in $\mathcal{Q}^*(\Omega)$ by

$$\mu(\varphi(A,B), \varphi(A',B')) = \begin{cases} (-1)^{|A \setminus A'| + |B \setminus B'|} & \text{if } \varphi(A,B) \subseteq \varphi(A',B'), \\ 0 & \text{otherwise.} \end{cases}$$

Proof: The proof is similar to that of Theorem 2 in [50] with simple adaptations, due to the similarity between two-place belief functions on $\mathcal{C}(\Omega)$ and bi-capacities (see Section 4.5 below). □

This result allows us to compute m from b as:

$$m(A,B) = \sum_{C \supseteq A, D \supseteq B} (-1)^{|C \setminus A| + |D \setminus B|} b(C,D), \tag{4.60}$$

and from q as

$$m(A,B) = \sum_{C \subseteq A, D \subseteq B} (-1)^{|A \setminus C| + |B \setminus D|} q(C,D). \tag{4.61}$$

The conjunctive and disjunctive weight functions may be computed, respectively, as:

$$w(A,B) = \prod_{C \subseteq A, D \subseteq B} q(C,D)^{(-1)^{|A \setminus C| + |B \setminus D| + 1}}, \quad \forall (A,B) \neq (\emptyset_\Omega, \emptyset_\Omega), \tag{4.62}$$

and

$$v(A,B) = \prod_{C \supseteq A, D \supseteq B} b(C,D)^{(-1)^{|C \setminus A| + |D \setminus B| + 1}}, \quad \forall (A,B) \neq (\Omega, \Omega), \tag{4.63}$$

which makes it possible to use the cautious and bold rules in this context.

Example 6 Let V now denote the set of languages spoken by Bernard. Assume that we are 100 % sure that Bernard speaks no other language than Dutch (d), English (e) and French (f), so that we can restrict the domain of V to $\Omega = \{d, e, f\}$. Suppose that we have the following items of evidence:

1. Bernard is Belgian. Approximately 60 % of Belgians are Dutch-speaking, and 40 % of Belgians are French-speaking (we neglect here the small German-speaking community for simplicity). According to a recent survey, approximately 20 % of French-speaking Belgians declare to have good knowledge of Dutch, whereas around 50 % of members of the Dutch speaking community claim to have good knowledge of French.

2. Bernard studied three years in Canada, where most universities are English-speaking, and some are French speaking. Based on available evidence, we have a 0.7 degree of belief that Bernard studied in an English-speaking university, and a 0.15 degree of belief that he studied in a French-speaking one.

Each of these two items of evidence can be represented by a mass function on $\mathcal{C}(\Omega)$. According to the first item of evidence, approximately $(0.6 \times 0.5) \times 100 = 30\%$ of Belgians speak Dutch and no French, approximately $(0.4 \times 0.8) \times 100 = 32\%$ speak French and no Dutch, and the rest speak both languages. Knowing that Bernard belongs to this

population (and nothing else), and assuming these figures to be accurate, this would lead to the following mass function:

$$m_1(\{d\},\{f\}) = 0.3, \quad m_1(\{f\},\{d\}) = 0.32, \quad m_1(\{f,d\},\emptyset) = 0.38.$$

To account for inaccuracy of these figures, we may *discount* this mass function [93] by transferring a fraction of the mass (say, 10%) to the greatest element of $\mathcal{C}(\Omega)$, i.e., $\varphi(\emptyset,\emptyset)$. We thus have

$$m_1(\{d\},\{f\}) = 0.3 \times 0.9 = 0.27, \quad m_1(\{f\},\{d\}) = 0.32 \times 0.9 \approx 0.29,$$

$$m_1(\{f,d\},\emptyset) = 0.38 \times 0.9 \approx 0.34, \quad m_1(\emptyset,\emptyset) = 0.1.$$

The second item of evidence can be represented by a mass function m_2 defined as:

$$m_2(\{e\},\emptyset) = 0.7, \quad m_2(\{f\},\emptyset) = 0.15, \quad m_2(\emptyset,\emptyset) = 0.15.$$

Assuming these two items of evidence to be distinct, they should be combined using the conjunctive sum operation $\textcircled{\cap}$. This may be achieved in two ways:

1. We may compute the intersection between each focal element of m_1 and each focal element of m_2 and apply formula (4.54). The computations may be presented as in Table 4.1.

2. Alternatively, we may compute the commonality functions q_1 and q_2 using (4.53), multiply them, and convert the result into a mass function using (4.61). The intermediate and final results are shown in Table 4.2.

We may check that both approaches yield the same result. In particular, we can see that the empty set $\emptyset_\Theta$ receives a mass equal to $0.15 \times 0.27 = 0.0405$, which can be interpreted as a degree of conflict between m_1 and m_2. Using the consensus rule $\boxdot$ (4.59), the mass 0.15×0.27 would be transferred to

$$\varphi(\{f\},\emptyset) \sqcap \varphi(\{d\},\{f\}) = \varphi(\{d\},\emptyset),$$

resulting in a normal, conflict-free mass function.

Table 4.2 also shows the normal mass function computed using the normalized Dempster's rule $\oplus$, and Table 4.3 displays the intermediate steps and final results for computing the combinations of m_1 and m_2 using the unnormalized and normalized cautious rules.

Table 4.1: Computation of the conjunctive sum of m_1 and m_2 in Example 6. The columns and the lines correspond to the focal elements of m_1, and m_2, respectively. Each cell contains the intersection of a focal element of m_1 and a focal element of m_2. The mass of each focal element is indicated below it.

	$(\{d\},\{f\})$ 0.27	$(\{f\},\{d\})$ 0.29	$(\{f,d\},\emptyset)$ 0.34	$(\emptyset,\emptyset)$ 0.1
$(\{e\},\emptyset)$ 0.7	$(\{d,e\},f)$ 0.7×0.27	$(\{e,f\},\{d\})$ 0.7×0.29	$(\{e,f,d\},\emptyset)$ 0.7×0.34	$(\{e\},\emptyset)$ 0.7×0.1
$(\{f\},\emptyset)$ 0.15	$\emptyset_\Theta$ 0.15×0.27	$(\{f\},\{d\})$ 0.15×0.29	$(\{f,d\},\emptyset)$ 0.15×0.34	$(\{f\},\emptyset)$ 0.15×0.1
$(\emptyset,\emptyset)$ 0.15	$(\{d\},\{f\})$ 0.15×0.27	$(\{f\},\{d\})$ 0.15×0.29	$(\{f,d\},\emptyset)$ 0.15×0.34	$(\emptyset,\emptyset)$ 0.15×0.1

Remark 2 We may remark here that the concept of two-place mass and belief functions defined here bears some similarity with bi-capacities introduced by Grabisch and Labreuche [50]. A bi-capacity as defined in [50] is an increasing function defined on the lattice $(\mathcal{Q}(\Omega), \sqsubseteq)$, where $\sqsubseteq$ is the partial ordering on $\mathcal{Q}(\Omega)$ defined by $(A,B) \sqsubseteq (C,D)$ if $A \subseteq B$ and $C \supseteq D$. In [50], Grabisch and Labreuche introduce various concepts related to bi-capacities, with application to cooperative game theory. In [66], they introduce the concept of bi-belief function, defined as a totally monotone bi-capacity from $\mathcal{Q}(\Omega)$ to $[0,1]$. They suggest an interpretation in terms of bipolar representation of uncertainty for the case of a single-valued variable. Bi-belief functions and two-place belief functions as introduced here are thus two distinct classes of belief functions built on different lattices, with different interpretations.

Remark 3 Another remark concerns decision making. As noted in the previous section, the lattice $(\mathcal{C}(\Omega), \subseteq)$ is not Boolean, so that the notion of pignistic probability cannot be defined in that lattice. In the classical setting, a common alternative to the rule of maximum pignistic probability for decision making is that of maximum plausibility: it consists in selecting the element of Ω with the greatest plausibility or, equivalent, with the greatest commonality (as these two functions coincide on singletons). In $\mathcal{C}(\Omega)$, we may propose as a reasonable decision rule to select the atom $\varphi(A, \overline{A})$ with the highest commonality. Table 4.4 shows the commonalities of the atoms computing from $m_1 \oplus m_2$, $m_1 \boxdot m_2$ and $m_1 \circledwedge^* m_2$ in Example 6. In that particular case, we can see that the three rules lead to the same conclusion, which is that Bernard speaks all three languages. The second most likely hypothesis is that Bernard speaks English and French, but no

Table 4.2: Computation of $m_1 \textcircled{\cap} m_2$ and $m_1 \oplus m_2$ in Example 6.

A	B	m_1	q_1	m_2	q_2	$q_{1\textcircled{\cap}2}$	$m_1 \textcircled{\cap} m_2$	$m_1 \oplus m_2$
$\{def\}$	$\{def\}$	0	1	0	1	1	0.0405	0
$\emptyset$	$\{def\}$	0	0.1	0	0.15	0.015	0	0
$\emptyset$	$\{de\}$	0	0.1	0	0.15	0.015	0	0
$\{f\}$	$\{de\}$	0	0.39	0	0.3	0.117	0	0
$\emptyset$	$\{df\}$	0	0.1	0	0.15	0.015	0	0
$\emptyset$	$\{d\}$	0	0.1	0	0.15	0.015	0	0
$\{f\}$	$\{d\}$	0.29	0.39	0	0.3	0.117	0.087	0.091
$\{e\}$	$\{df\}$	0	0.1	0	0.85	0.085	0	0
$\{e\}$	$\{d\}$	0	0.1	0	0.85	0.085	0	0
$\{ef\}$	$\{d\}$	0	0.39	0	1	0.39	0.203	0.212
$\emptyset$	$\{ef\}$	0	0.1	0	0.15	0.015	0	0
$\emptyset$	$\{e\}$	0	0.1	0	0.15	0.015	0	0
$\{f\}$	$\{e\}$	0	0.1	0	0.3	0.03	0	0
$\emptyset$	$\{f\}$	0	0.1	0	0.15	0.015	0	0
$\emptyset$	$\emptyset$	0.1	0.1	0.15	0.15	0.015	0.015	0.016
$\{f\}$	$\emptyset$	0	0.1	0.15	0.3	0.03	0.015	0.016
$\{e\}$	$\{f\}$	0	0.1	0	0.85	0.085	0	0
$\{e\}$	$\emptyset$	0	0.1	0.7	0.85	0.085	0.07	0.07
$\{ef\}$	$\emptyset$	0	0.1	0	1	0.1	0	0
$\{d\}$	$\{ef\}$	0	0.37	0	0.15	0.0555	0	0
$\{d\}$	$\{e\}$	0	0.1	0	0.15	0.015	0	0
$\{df\}$	$\{e\}$	0	0.44	0	0.3	0.132	0	0
$\{d\}$	$\{f\}$	0.27	0.37	0	0.15	0.0555	0.0405	0.0422
$\{d\}$	$\emptyset$	0	0.1	0	0.15	0.015	0	0
$\{df\}$	$\emptyset$	0.34	0.44	0	0.3	0.132	0.102	0.106
$\{de\}$	$\{f\}$	0	0.37	0	0.85	0.3145	0.189	0.197
$\{de\}$	$\emptyset$	0	0.1	0	0.85	0.085	0	0
$\{def\}$	$\emptyset$	0	0.44	0	1	0.44	0.238	0.248

Table 4.3: Computation of $m_1 \textcircled{\wedge} m_2$ and $m_1 \textcircled{\wedge}^* m_2$ in Example 6.

A	B	m_1	w_1	m_2	w_2	$w_{1\wedge 2}$	$m_1 \textcircled{\wedge} m_2$	$m_1 \textcircled{\wedge}^* m_2$
$\{def\}$	$\{def\}$	0	6.349	0	1	1	0.864	0
$\emptyset$	$\{def\}$	0	1	0	1	1	0	0
$\emptyset$	$\{de\}$	0	1	0	1	1	0	0
$\{f\}$	$\{de\}$	0	1	0	1	1	0	0
$\emptyset$	$\{df\}$	0	1	0	1	1	0	0
$\emptyset$	$\{d\}$	0	1	0	1	1	0	0
$\{f\}$	$\{d\}$	0.29	0.256	0	1	0.256	0.00806	0.0591
$\{e\}$	$\{df\}$	0	1	0	1	1	0	0
$\{e\}$	$\{d\}$	0	1	0	1	1	0	0
$\{ef\}$	$\{d\}$	0	1	0	1	1	0.0376	0.276
$\emptyset$	$\{ef\}$	0	1	0	1	1	0	0
$\emptyset$	$\{e\}$	0	1	0	1	1	0	0
$\{f\}$	$\{e\}$	0	1	0	1	1	0	0
$\emptyset$	$\{f\}$	0	1	0	1	1	0	0
$\emptyset$	$\emptyset$	0.1	10	0.15	6.67	719.6	0.00139	0.0102
$\{f\}$	$\emptyset$	0	1	0.15	0.5	0.5	0.00139	0.0102
$\{e\}$	$\{f\}$	0	1	0	1	1	0	0
$\{e\}$	$\emptyset$	0	1	0.7	0.176	0.176	0.00649	0.0477
$\{ef\}$	$\emptyset$	0	1	0	1.7	1	0.00649	0.0477
$\{d\}$	$\{ef\}$	0	1	0	1	1	0	0
$\{d\}$	$\{e\}$	0	1	0	1	1	0	0
$\{df\}$	$\{e\}$	0	1	0	1	1	0	0
$\{d\}$	$\{f\}$	0.27	0.270	0	1	0.270	0.00375	0.0275
$\{d\}$	$\emptyset$	0	1	0	1	1	0	0
$\{df\}$	$\emptyset$	0.34	0.227	0	1	0.227	0.00945	0.0694
$\{de\}$	$\{f\}$	0	1	0	1	1	0.0175	0.129
$\{de\}$	$\emptyset$	0	1	0	1	1	0	0
$\{def\}$	$\emptyset$	0	1	0	1	1	0.0441	0.324

Table 4.4: Commonalities of atoms according to $m_1 \oplus m_2$, $m_1 \boxdot m_2$ and $m_1 \circledwedge^* m_2$ in Example 6.

$(A, \overline{A})$	$q_{1\oplus 2}(A, \overline{A})$	$q_{1\boxdot 2}(A, \overline{A})$	$q_{1\wedge^* 2}(A, \overline{A})$
$(\emptyset, \{def\})$	0.0156	0.015	0.0102
$(\{f\}, \{de\})$	0.122	0.117	0.0796
$(\{e\}, \{df\})$	0.0889	0.085	0.0578
$(\{ef\}, \{d\})$	0.406	0.39	0.451
$(\{d\}, \{ef\})$	0.0578	0.096	0.0377
$(\{df\}, \{e\})$	0.138	0.173	0.0898
$(\{de\}, \{f\})$	0.328	0.355	0.214
$\boldsymbol{(\{def\}, \emptyset)}$	**0.459**	**0.481**	**0.509**

Dutch. However, it is clear that different combination rules may, in general, result in different decisions.

The following section will be devoted to a review of previous work on uncertainty representation for set-valued variables.

4.5 Relation to Previous Work

This section discusses the relation between the notions introduced above and related concepts or other formalisms already proposed for handling set-valued variables.

4.5.1 Disjunctive vs. Conjunctive Bodies of Evidence

Yager [115][116] was among the first authors to emphasize the fundamental difference between single-valued and set-values variables, and to develop specific formalisms for reasoning with the latter. In [116], a distinction is made between *disjunctive* and *conjunctive* information using set-based representations. Given a variable V taking a single value in Ω, a statement "V is A" with $A \subseteq \Omega$ means that V takes *some* value in A, but we do not know which one. In contrast, if V is multiple-valued, the same statement is understood to mean that V takes *all* values in A (and possibly other values outside A). The corresponding piece of information is called "disjunctive" in the former case, and "conjunctive" in the latter. Yager then proceeds by observing that there is some kind of duality between disjunctive and conjunctive knowledge. For instance, the statement P_1 : "V is A" implies P_2 : "V is B" whenever $B \supseteq A$ in the disjunctive case, whereas P_2 can be deduced from P_1 whenever $B \subseteq A$ in the conjunctive case. If we know that P_1 and P_2 both hold, then we can deduce "V is $A \cap B$" in the disjunctive case, and "V is $A \cup B$" in the conjunctive case, etc.

Viewing mass functions as generalized sets, Dubois and Prade [32] remarked that the same distinction holds in the belief function framework. They pointed out that, when a mass function m represents a body of evidence pertaining to a set-valued variable (referred to as a conjunctive body of evidence), the commonality function q is more appropriate than b for representing degrees of belief, and the disjunctive sum (4.11) should be used for combining information conjunctively.

The formalism developed in Section 4.4 sheds new light on this duality between conjunctive and disjunctive knowledge. The conjunctive statement "V is A" corresponds to the proposition $\varphi(A, \emptyset)$. Let m be a mass function on $\mathcal{C}(\Omega)$ whose focal elements are all of the form $\varphi(B, \emptyset)$ for some $B \subseteq \Omega$. We can note $m'(A) = m(A, \emptyset)$ for all A. Using (4.51), we then have, for all $A \subseteq \Omega$:

$$b(A, \emptyset) = \sum_{B \supseteq A} m(B, \emptyset) = \sum_{B \supseteq A} m'(B) = q'(A),$$

where q' is the commonality function corresponding to m'. Conversely,

$$q(A, \emptyset) = \sum_{B \subseteq A} m(B, \emptyset) = \sum_{B \subseteq A} m'(B) = b'(A).$$

As a consequence, let m_1 and m_2 be two mass functions on $\mathcal{C}(\Omega)$ with focal elements of the form described above, and assume that we want to combine them conjunctively using (4.56). We get

$$q_{1 \textcircled{\cap} 2}(A, \emptyset) = q_1(A, \emptyset) q_2(A, \emptyset) = b'_1(A) b'_2(A) = b'_{1 \textcircled{\cup} 2}(A)$$

for all $A \subseteq \Omega$, which explains why the disjunctive sum *seems* to be used when combining conjunctive bodies of evidence in a conjunctive manner.

4.5.2 Random sets

Random sets are defined as random elements taking values as subsets of some space [73][82]. In the finite case, a random set is thus defined by a probability function m on 2^Ω such that $\sum_{A \subseteq \Omega} m(A) = 1$, which is mathematically equivalent to a Dempster-Shafer mass function on Ω [81]. However, as noted by Smets [96], the semantics of random sets and (standard) belief functions are different, as random sets model random experiments with set-valued outcomes, whereas standard belief functions quantify beliefs regarding a variable taking a single, but unknown value.

In contrast, random sets are recovered as a special class of belief functions on set-valued variables introduced in this chapter. Let m be a mass function on $\mathcal{C}(\Omega)$, and assume that the focal elements of m are atoms of $\mathcal{C}(\Omega)$, i.e., if they are of the form $(A, \overline{A})$. In that case, the function m' from 2^Ω to $[0, 1]$ such that $m'(A) = m(A, \overline{A})$ for all $A \subseteq \Omega$ is a random set. Random sets are thus equivalent to mass functions on $\mathcal{C}(\Omega)$ with atomic focal elements, just as probability distributions on Ω are equivalent to mass functions on 2^Ω with singleton focal elements.

4.5.3 Veristic Variables

In Chapter 3 we have presented the theory of veristic variables proposed by Yager in [119][117]. As we have already mentioned, a veristic variable can be viewed as a fuzzy set-valued variables. Let V denote such variable defined over Ω. Clearly, a major difference between Yager's approach and ours is the fact that Yager represents each piece

of knowledge about V as a set of *fuzzy* subsets of Ω, whereas we use a set of *crisp* subsets of Ω. However, the kinds of statements considered by Yager as well as the associated verity and rebuff distributions have very close representations in our approach.

To begin with, let us provisionally assume that A is a crisp subset of Ω. Then, each of the four types of statements, already introduced in Section 3.3, can be expressed by categorical mass functions on $\mathcal{C}(\Omega)$ as follows:

$$\begin{array}{rcl} V\ isv\ A & \longrightarrow & m(A,\emptyset)=1 \\ V\ isv(n)\ A & \longrightarrow & m(\emptyset,A)=1 \\ V\ isv(c)\ A & \longrightarrow & m(A,\overline{A})=1. \\ V\ isv(c,n)\ \overline{A} & \longrightarrow & m(\overline{A},A)=1. \end{array}$$

It is easy to see that, in each of these four cases:

$$b(\{\omega\},\emptyset) = \text{Ver}(\omega) \tag{4.64}$$

$$b(\emptyset,\{\omega\}) = \text{Rebuff}(\omega) \tag{4.65}$$

for all $\omega \in \Omega$. The verity of ω is thus the belief that ω is one of the values taken by V, whereas the rebuff of ω is the belief that ω is not a value taken by V. This interpretation can be shown to remain true when A is a fuzzy subset of Ω. In that case, the function $\omega \rightarrow A(\omega)$ can be seen as a possibility distribution, which is known to be equivalent to a consonant mass function m' on Ω with focal elements $A_1 \subseteq \ldots \subseteq A_n$. The corresponding plausibility function pl' verifies

$$pl'(\{\omega\}) = \sum_{A_i \ni \omega} m'(A_i) = A(\omega), \quad \forall \omega \in \Omega.$$

For instance, let us consider the statement $V\ isv\ A$, and let us translate it as the following two-place mass function:

$$m(A_i,\emptyset) = m'(A_i), \quad i = 1,\ldots,n.$$

We have

$$b(\{\omega\},\emptyset) = \sum_{A_i \ni \omega} m(A_i,\emptyset) = \sum_{A_i \ni \omega} m'(A_i) = A(\omega) = \text{Ver}(\omega)$$

and

$$b(\emptyset,\{\omega\}) = 0 = \text{Rebuff}(\omega).$$

By handling the three other cases similarly, it can be verified that Equations (4.64) and (4.65) hold in all cases.

We may thus conclude that, although based on a slightly different interpretation, Yager's framework can be easily translated into the formalism of two-place belief functions, which is more general. However, this is only true at the *static* level, i.e., as long as we do not combine different pieces of information. For instance, as shown by Yager, the conjunctive combination of two statements V *isv* A and V *isv* B in the veristic framework results in a new statement V *isv* $A \cup B$, where $\cup$ denotes fuzzy set union. This is consistent with our approach only as long as A and B are crisp sets. If A and B are fuzzy, then translating the two statements as two-place mass functions and combining them using either the conjunctive sum or the cautious rule does not, in general, yield a consonant mass function corresponding to a veristic constraint on V. The two formalisms thus differ when combining statements involving fuzzy subsets.

4.5.4 Two-fold fuzzy sets

To complete this review of previous work on uncertainty representation for set-valued variables, we need to mention the representation of incomplete conjunctive information using a pair of fuzzy sets introduced in [27].

In this work, Dubois and Prade proposed to represent partial knowledge about a set-valued variable as a possibility distribution π on 2^Ω. This is equivalent to defining a fuzzy set of crisp subsets of Ω, which contrasts with Yager's approach who defines a crisp set W of fuzzy subsets of Ω. To make such a representation more easily tractable, Dubois and Prade then proposed to approximate π by a pair of fuzzy sets (A^-, A^+) as follows. Let A_i, $i \in I$ be the family of subsets of Ω such that $\pi(A_i) > 0$. Let

$$A^-(\omega) = 1 - \sup_{i:\omega \notin A_i} \pi(A_i)$$

and

$$A^+(x) = \sup_{i:\omega \in A_i} \pi(A_i).$$

The degree of membership of V to A^- is thus the extent to which is impossible to find an A_i not containing V, while $A^+(x)$ corresponds to the possibility of finding an A_i containing V. The pair (A^-, A^+), referred to as a *two-fold fuzzy set*, constitutes an approximation of π in the sense that it is a simpler, but incomplete representation: several possibility distributions π correspond to the same two-fold fuzzy set. However, Dubois and Prade showed that the least specific possibility distribution π^* induced by

a two-fold fuzzy set (A^-, A^+) can be expressed as $\pi^*(\emptyset) = 1 - \sup A^-$, $\pi^*(\Omega) = \inf A^+$, and

$$\pi^*(B) = \min\left[\inf_{x\in B} A^+(\omega), \inf_{\omega\notin B}(1 - A^-(\omega))\right], \quad \forall B \in 2^\Omega \setminus \{\emptyset, \Omega\}.$$

To each two-fold fuzzy set (A^-, A^+) can thus be associated a fuzzy subset $\mathcal{A}$ of 2^Ω, with membership function equal to π^*.

We note that this approach has some similarity with ours, since it is based on the representation of a subset of 2^Ω by a pair of subsets of Ω. Actually, if A^- and A^+ are crisp, then the corresponding crisp subset $\mathcal{A}$ of 2^Ω is exactly equal to $\varphi(A^-, \overline{A^+})$. However, in the general case, the two-fold fuzzy set representation is based on a pair of possibility distributions, i.e., consonant belief functions on Ω, whereas our approach is based on a single two-place belief function on $\mathcal{C}(\Omega)$.

What can be seen as a limitation of the two-fold fuzzy set approach arises when combining information from several sources. Given two pairs (A^-, A^+) and (B^-, B^+) representing knowledge about two set-valued variables V_1 and V_2, Dubois and Prade showed that the knowledge of $V_1 \cap V_2$ can be represented by $(A^- \cap B^-, A^+ \cap B^+)$, while the knowledge of $V_1 \cup V_2$ can be represented by $(A^- \cup B^-, A^+ \cup B^+)$. Applications of this kind of reasoning to database query evaluation is discussed in [27]. However, a different and maybe more common problem in uncertain reasoning is the situation where we have two items of evidence about a single set-valued variable V, and we want to combine these two items of evidence. If (A^-, A^+) and (B^-, B^+) correspond, respectively, to fuzzy subsets $\mathcal{A}$ and $\mathcal{B}$ of 2^Ω, the result of the combination should ideally correspond to $\mathcal{A} \cap \mathcal{B}$ or to $\mathcal{A} \cup \mathcal{B}$, depending on the choice of a conjunctive or disjunctive combination mechanism. However, none of these two fuzzy subsets of 2^Ω generally admits a two-fold fuzzy set representation, which restricts the use of this formalism for reasoning with set-valued variables.

We have shown that the formalism of two-place belief functions introduced in this chapter seems to compare favorably in terms of expressive power with existing formalisms for representing and reasoning with uncertain conjunctive information. In the next section, we will demonstrate the usefulness of this formalism for multi-label classification problems.

4.6 Application to Multi-label Classification

In this section, we present an application of the proposed two-place belief functions framework to multi-label classification.

The class label of each instance may be considered as a set-valued variable. As remarked in Section 3.4.1, in order to construct a multi-label classifier, we generally assume the existence of a labeled training set where each instance $\mathbf{x}_i$ is assigned a *single* subset Y_i of the set $\mathcal{Y}$ of classes. In practice, however, gathering such high quality information is not always possible, especially, when the instances have been labeled *subjectively* by one or several experts. Uncertainty may be introduced in the labeling process, and thus, it will be very difficult to *precisely* label each instance.

For example, assume that instances are songs and classes are emotions generated by these songs, as in the emotion dataset that will be used later in the experiments. Upon hearing a song, an expert may decide that this song certainly evokes happiness and certainly does not evoke sadness, but may be undecided regarding the other emotions (such as quietness, anger, surprise, etc.). In that case, the song cannot be assigned a single label set, but we can associate to it the set of all label sets containing "happiness" and not containing "sadness", which has the form suggested above.

The formalism developed in this may can easily be used to handle such situations. In the most general setting, the opinions of one or several experts regarding the set of classes that pertain to a particular instance $\mathbf{x}_i$ may be modeled by a mass function m_i on $\mathcal{C}(\Omega)$. A less general, but arguably more workable option is to restrict m_i to be categorical, i.e., to have a single focal element $\varphi(A_i, B_i)$, with $A_i, B_i \subseteq \Omega$ and $A_i \cap B_i = \emptyset$. The set A_i is then the set of classes that *certainly apply* to $\mathbf{x}_i$, while B_i the set of classes that certainly *do not* apply. When data are labeled by several experts, A_i might represent the set of classes indicated by all (or most) experts as relevant to describe instance $\mathbf{x}_i$, while B_i would be the set of classes mentioned by none of the experts (or only a few of them). The usual situation of precise labeling is recovered in the special case where $B_i = \overline{A}_i$.

In [20][132], we introduced a single-label k-nearest neighbor (NN) classifier based on Dempster-Shafer theory. This method will be briefly recalled in Subsection 4.6.1, and will be extended to multi-label classification tasks in Subsection 4.6.2. The proposed method is called EMLkNN, for Evidential Multi-Label k-NN.

4.6.1 Single-label Evidential k-NN Classification

The evidential k-NN method introduced in [20] for single-label classification problems can be summarized as follows. Let $\mathcal{D} = \{(\mathbf{x}_1, A_1), \ldots, (\mathbf{x}_n, A_n)\}$ be a learning set of n instances, where $\mathbf{x}_i$ belongs to the domain of instances $\mathbb{X}$ and $A_i \subseteq \mathcal{Y}$ is a set of possible classes for this instance. We emphasize the fact that, in the context considered here, each instance $\mathbf{x}_i$ actually belongs to one and only class, but this class is only known to lie somewhere in A_i.

Let $\mathbf{x}$ be an new instance that we search to estimate its class y. We want to guess the value of y based on evidence provided by the learning set $\mathcal{D}$. For that purpose, we consider the set $\mathcal{N}_{\mathbf{x}}^k$ of the k nearest neighbors of $\mathbf{x}$, according to some distance measure d. Each learning object $(\mathbf{x}_i, A_i)$ with $\mathbf{x}_i \in \mathcal{N}_{\mathbf{x}}^k$ can then be regarded as a piece of evidence regarding the unknown value of y, represented as the following simple mass function on $\mathcal{Y}$:

$$m_i(A_i) = \alpha_0 \exp\left(-\gamma d(\mathbf{x}, \mathbf{x}_i)\right), \tag{4.66}$$

$$m_i(\Omega) = 1 - \alpha_0 \exp\left(-\gamma d(\mathbf{x}, \mathbf{x}_i)\right), \tag{4.67}$$

with $0 < \alpha_0 < 1$ and $\gamma > 0$. Parameter α_0 is usually fixed at a value close to 1 such as $\alpha_0 = 0.95$, whereas γ should depend on the scaling of distances and can be either fixed heuristically or optimized [132]. We recall that the same function $\alpha_0 \exp(-\gamma d(\mathbf{x}, \mathbf{x}_i)$ was used for knowledge discounting in the VERkNN method. The evidence of the k NNs is then pooled using the conjunctive sum:

$$m = \textcircled{\cap}_{i:\mathbf{x}_i \in \mathcal{N}_{\mathbf{x}}^k} m_i, \tag{4.68}$$

and the class with highest plausibility or pignistic probability is selected. As remarked in [23] and [22], this method can be easily extended to the case where each learning instance in $\mathcal{D}$ is labeled by a general mass function on $\mathcal{Y}$.

4.6.2 Multi-label Evidential k-NN Classification

Let us now come back to the multi-label classification problem, in which objects may belong *simultaneously* to several classes. Let $\mathcal{D} = \{(\mathbf{x}_1, A_1, B_1), \ldots, (\mathbf{x}_n, A_n, B_n)\}$ be the learning set, where $A_i \subseteq \mathcal{Y}$ denotes a set of classes that surely apply to the instance $\mathbf{x}_i$, and $B_i \subseteq \Omega$ a set of classes that surely do not apply to the same instance. If $Y_i \subseteq \mathcal{Y}$

denotes the true label set of $\mathbf{x}_i$, we thus only know that $Y_i \in \varphi(A_i, B_i)$. The EML$k$NN method builds a multi-label classifier $\mathcal{H}$ and a scoring function f as it will be explained in the following.

As before, let $\mathcal{N}_{\mathbf{x}}^k$ denote the set of k nearest neighbors of a new instance $\mathbf{x}$, and $\mathbf{x}_i$ an element of that set with label (A_i, B_i). This item of evidence can be described by the following simple two-valued mass function:

$$m_i(A_i, B_i) = \alpha \exp\left(-\gamma d(\mathbf{x}, \mathbf{x}_i)\right), \tag{4.69}$$

$$m_i(\emptyset, \emptyset) = 1 - \alpha \exp\left(-\gamma d(\mathbf{x}, \mathbf{x}_i)\right), \tag{4.70}$$

with, as before, $0 < \alpha_0 < 1$ and $\gamma > 0$. These k mass functions are then combined using a combination rule (the conjunctive sum, the consensus or the cautious rules).

For decision making, different procedures can be used. The following simple and computationally efficient rule was implemented. To decide whether to assign each class $\omega \in \mathcal{Y}$ or not to instance $\mathbf{x}$, we compute the degree of belief $bel(\{\omega\}, \emptyset)$ that the true label set Y contains ω, and the degree of belief $bel(\emptyset, \{\omega\})$ that it does not contain ω. We then define the multi-label classifier $\mathcal{H}$ as

$$\mathcal{H}(\mathbf{x}) = \{\omega \in \mathcal{Y} \mid bel(\{\omega\}, \emptyset) \geq bel(\emptyset, \{\omega\})\},$$

and the corresponding scoring function f as

$$f(\mathbf{x}, \omega) = bel(\{\omega\}, \emptyset).$$

4.7 Conclusion

We have presented a formalism for quantifying uncertainty on a set-valued variable V defined on a domain Ω in the belief function framework. This approach relies on the definition of a family $\mathcal{C}(\Omega)$ of subsets of 2^Ω that is closed under intersection and has a lattice structure. Each element in $\mathcal{C}(\Omega)$ is indexed by two subsets A and B, and is defined as the set of subsets of Ω containing A and not intersecting B. The number of such elements (including the empty set of 2^Ω) is equal to $3^{|\Omega|} + 1$: it is thus much smaller than the size of 2^{2^Ω}, while being rich enough to express evidence about set-valued variables in many realistic situations.

Most notions from Dempster-Shafer theory of belief functions can be defined on $\mathcal{C}(\Omega)$. The proposed formalism has been shown to be somewhat similar to, but arguably more general and flexible than other approaches introduced in the possibilistic framework.

Finally, based on the proposed two-place belief functions framework formalism, we have proposed a multi-label classification method, called EMLkNN, where each unseen instance is classified on the basis of its k nearest neighbors. In particular, the EMLkNN method allows us to handle multi-label learning problems with imprecise labels.

Chapter 5

Experiments

Summary

In this chapter, we show a comparison between the proposed methods and with other state-of-the-art multi-label learning algorithms on several benchmark datasets and using different evaluation criteria. We report experimental results on both precisely and imperfectly labeled data. The latter case occurs when, for example, the data have been labeled subjectively by one or many experts in the absence of ground truth. Due to lack of confidence and conflits between experts, noisy and imprecise labels will inevitably be introduced in the labeling process.

Résumé

Dans ce chapitre, nous montrons une comparaison, sur plusieurs jeux de données et en utilisant différents critères d'évaluation, entre les méthodes proposées et avec d'autres algorithmes de l'état de l'art de l'apprentissage multi-label. Nous présentons des résultats expérimentaux sur des données étiquetées d'une façon précise en premier lieu, et d'une façon imparfaite en deuxième lieu. En fait, il n'est pas toujours possible de disposer de données qui sont parfaitement étiquetées. En effet, dans de nombreuses applications réelles, il n'existe pas de vérité terrain pour l'étiquetage des différents individus sans aucune ambiguïté, et plusieurs experts doivent être consultés. En raison de conflits entre les experts et de manque d'informations, des imprécisions et des bruits seront introduits durant l'étiquetage des données.

5.1 Introduction

Three methods for multi-label learning have been proposed in this thesis: DMLkNN, VERkNN and EMLkNN. We present in this chapter a comparative study between the proposed methods and with some state-of-the-art algorithms using several benchmark datasets and different multi-label evaluation measures.

In order to construct a multi-label classifier, we generally assume the existence of a labeled training set in which each instance is assigned a *precise* set of labels. In practice, however, gathering such high quality information is not always feasible at a reasonable cost. In many problems, there is no ground truth for assigning unambiguously a label set to each instance, and the opinions of one or several expert have to be elicited. Typically, an expert will sometimes express lack of confidence for assigning a well-known label set. If several experts are consulted, some conflict will inevitably arise, which again will introduce some uncertainty in the labeling process, and lead to imperfect labeled data. The experimental study presented below address both cases of precise and imperfect labeled data.

This chapter is organized as follow. Section 5.2 will present the evaluation metrics used for the comparison of the different methods. The benchmark datasets used in our experiments will be reported in Section 5.3. Experimental results on precise data will be detailed in Section 5.4, and Section 5.5 will present a comparative study on imperfect labeled data. Finally,some concluding remarks will be made in Section 5.6.

5.2 Evaluation metrics

The evaluation of multi-label learning systems is more complex from that of single-label learning systems. A result can be fully correct, partially correct or fully wrong. Let $\mathcal{H} : \mathbb{X} \longrightarrow 2^{\mathcal{Y}}$ be a multi-label classifier that assigns a subset of $\mathcal{Y} = \{\omega_1, \ldots, \omega_Q\}$ for each instance $\mathbf{x} \in \mathbb{X}$, and let $f : \mathbb{X} \times \mathcal{Y} \longrightarrow [0, 1]$ be the corresponding scoring function that attributes a score to each class $\omega_q \in Y$ interpreted as the probability that $\mathbf{x}$ belongs to ω_q. There exist a number of evaluation criteria that evaluate the performance of a multi-label learning system, given a set $D = \{(\mathbf{x}_1, Y_1), \ldots, (\mathbf{x}_n, Y_n)\}$ of test examples. We give hereafter some of the main evaluation criteria used in the literature to evaluate a multi-label learning system [91][102]. The evaluation metrics can be divided into two groups: *prediction-based* and *ranking-based* metrics. Prediction-based metrics assess

the *correctness* of the label sets predicted by the multi-label classifier $\mathcal{H}$, while ranking-based metrics evaluate the label ranking quality depending on the scoring function f. As a scoring function is not computed by all multi-label classification methods, the former category of metrics is of more general use.

5.2.1 Prediction-based metrics

Accuracy The accuracy metric is an average degree of similarity between the predicted and the ground truth label sets of all test examples:

$$Acc(\mathcal{H}, \mathcal{S}) = \frac{1}{n} \sum_{i=1}^{n} \frac{|Y_i \cap \widehat{Y}_i|}{|Y_i \cup \widehat{Y}_i|}.$$

where $\widehat{Y}_i = \mathcal{H}(\mathbf{x}_i)$ denotes the predicted label set of instance $\mathbf{x}_i$.

F1-measure The F1-measure is defined as the harmonic mean of two other metrics called Precision ($Prec$) and Recall (Rec) [121]. The former computes the proportion of correct positive predictions while the latter calculates the proportion of true labels that have been predicted as positives. These metrics are defined as follow:

$$Prec(\mathcal{H}, \mathcal{S}) = \frac{1}{n} \sum_{i=1}^{n} \frac{|Y_i \cap \widehat{Y}_i|}{|\widehat{Y}_i|},$$

$$Rec(\mathcal{H}, \mathcal{S}) = \frac{1}{n} \sum_{i=1}^{n} \frac{|Y_i \cap \widehat{Y}_i|}{|Y_i|},$$

and,

$$F1(\mathcal{H}, \mathcal{S}) = \frac{2 \cdot Prec \cdot Rec}{Prec + Rec} = \frac{1}{n} \sum_{i=1}^{n} \frac{2|Y_i \cap \widehat{Y}_i|}{|Y_i| + |\widehat{Y}_i|}.$$

Hamming loss This metric counts prediction errors (an incorrect label is predicted) and missing errors (a true label is not predicted):

$$HLoss(\mathcal{H}, \mathcal{S}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{Q} |Y_i \Delta \widehat{Y}_i|,$$

where Δ stands for the symmetric difference between two sets.

Note that the values of the prediction-based evaluation criteria are in the interval $[0, 1]$. Larger values of the first four metrics correspond to higher classification quality, while for the Hamming loss metric, the smaller the symmetric difference between predicted and true label sets, the better the performance [102][121].

5.2.2 Ranking-based metrics

As stated before, this group of criteria is based on the scoring function $f(.,.)$ and evaluates the ranking quality of the different possible labels [39][129]. Let $rank_f(.,.)$ be the ranking function derived from f and taking values in $\{1,\ldots,\mathrm{Q}\}$. For each instance $\mathbf{x}_i$, the label with the highest scoring value has rank 1, and if $f(\mathbf{x}_i,\omega_q) > f(\mathbf{x}_i,\omega_r)$, then $rank_f(\mathbf{x}_i,\omega_q) < rank_f(\mathbf{x}_i,\omega_r)$.

One-error The one-error metric evaluates how many times the top-ranked label, i.e. the label with the highest score, is not in the true set of labels of the instance:

$$OErr(f,\mathcal{S}) = \frac{1}{n}\sum_{i=1}^{n}\langle[\arg\max_{\omega\in Y} f(\mathbf{x}_i,\omega)] \notin Y_i\rangle,$$

where for any proposition H, $\langle H\rangle$ equals to 1 if H holds and 0 otherwise. Note that, for single-label classification problems, the one-error is identical to ordinary classification error.

Coverage The coverage measure is defined as the average number of steps needed to move down the ranked label list in order to cover all the labels assigned to a test instance:

$$Cov(f,\mathcal{S}) = \frac{1}{n}\sum_{i=1}^{n}\max_{\omega\in Y_i} rank_f(\mathbf{x}_i,\omega) - 1.$$

Ranking loss This metric calculates the average fraction of label pairs that are reversely ordered for an instance:

$$RLoss(f,\mathcal{S}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i||\overline{Y}_i|}|\{(\omega_q,\omega_r)\in Y_i\times\overline{Y}_i \mid f(\mathbf{x}_i,\omega_q)\leq f(\mathbf{x}_i,\omega_r)\}|$$

where $\overline{Y}_i$ denotes the complement of Y_i in Y.

Average precision This criteria was first used in information retrieval and was then adapted to multi-label learning problems in order to evaluate the effectiveness of label ranking. This metric measures the average fraction of labels ranked above a particular label $y\in Y_i$ which actually are in Y_i:

$$AvPrec(f,\mathcal{S}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|}\sum_{\omega_q\in Y_i}\frac{|\{\omega_r\in Y_i \mid rank_f(\mathbf{x}_i,\omega_r)\leq rank_f(\mathbf{x}_i,\omega_q)\}|}{rank_f(\mathbf{x}_i,\omega_q)}.$$

For the ranking-based metrics, smaller values of the first three metrics correspond to better label ranking quality, while $AvPrec(f, \mathcal{S}) = 1$ means that the labels are perfectly ranked for all test examples [39].

5.3 Multi-labeled datasets

5.3.1 Multi-label Statistics

Given a multi-labeled dataset $\mathcal{D} = \{(\mathbf{x}_i, Y_i), i = 1, \ldots, n\}$ with $\mathbf{x}_i \in \mathbb{X}$ and $Y_i \subseteq \mathcal{Y}$, the following measures give some statistics about the "label multiplicity" of the dataset $\mathcal{D}$ [102].

- The *label cardinality* of $\mathcal{D}$, denoted by $LCard(\mathcal{D})$, indicates the average number of labels per instance:
$$LCard(\mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} |Y_i|$$
- The *label density* of $\mathcal{D}$, denoted by $LDen(\mathcal{D})$, is defined as the average number of labels per instance divided by the number of possible labels Q:
$$LDen(\mathcal{D}) = \frac{LCard(\mathcal{D})}{\mathrm{Q}}$$
- $DL(\mathcal{D})$ counts the number of *distinct label sets* appeared in the dataset $\mathcal{D}$:
$$DL(\mathcal{D}) = |\{Y_i \subseteq \mathcal{Y} | \exists\, \mathbf{x}_i \in \mathbb{X} \ : \ (\mathbf{x}_i, Y_i) \in \mathcal{D}\}|$$

5.3.2 Benchmark datasets

Several real datasets[1] were used in our experiments. The used datasets come from different domains of application: text categorization, bioinformatics, and multimedia applications (music and image).

- The *emotion dataset*, presented in [101], consists of 593 songs annotated by experts according to the emotions they generate. The emotions are: amazed-surprise, happy-pleased, relaxing-calm, quiet-still, sad-lonely and angry-fearful. Each emotion corresponds to a class. There are thus 6 classes, and each song was labeled as

[1] These datasets can be downloaded from `http://mlkd.csd.auth.gr/multilabel.html`.

belonging to one or several classes. Each song was also described by 8 rhythmic features and 64 timbre features, resulting in a total of 72 features. The number of distinct label sets is equal to 27, the label cardinality is 1.868, and the label density is 0.311.

- The *scene dataset* consists of 2407 natural scene images. For each image, spatial color moments are used as features. Images are divided into 49 blocks using a 7×7 grid. The mean and variance of each band are computed corresponding to a low-resolution image and to computationally inexpensive texture features, respectively [8]. Each image is then transformed into a $49 \times 3 \times 2 = 294$-dimensional feature vector. A label set is manually assigned to each image. There are 6 different semantic scenes: *sea, sunset, trees, desert and mountains*. The average number of labels per instance is 1.074, thus the label density is 0.179 (only 7.35% of training instances are labeled by more than one class). The number of distinct sets of labels is equal to 15.

- The *yeast dataset* contains data regarding the gene functional classes of the yeast *Saccharomyces cerevisiae* [39]. It includes 2417 genes each represented by 103 features. Each gene is described by the concatenation of micro-array expression data and phylogenetic profile and is associated with a set of functional classes. There are 14 possible classes and there exist 198 distinct label combinations. The label cardinality is 4.237, and the label density is 0.303.

- The *medical dataset* consists of 978 examples each one represented by 1449 features. It is issued from the *Computational Medicine Center* concerning a challenge task on the automated processing of clinical free text. This dataset has been used in [89]. The average cardinality is 1.245, and the label density is 0.028 with 94 distinct label sets.

- The *Enron email dataset* consists of 1702 examples each one represented by 1001 features. It corresponds to messages belonging to users, mostly senior management of the *Enron Corp*. This dataset has been used in [89]. 753 distinct label combinations exist in the dataset. The label cardinality is 3.378 and the label density is 0.064.

Table 5.1: Characteristics of datasets

Dataset	Domain	Number of instances	Feature vector dimension	Number of labels	Label cardinality	Label density	Distinct label sets
emotion	music	593	72	6	1.868	0.311	27
scene	image	2407	294	6	1.074	0.179	15
yeast	biology	2417	103	14	4.237	0.303	198
medical	text	978	1449	45	1.245	0.028	94
enron	text	1702	1001	53	3.378	0.064	753

Table 5.2: Characteristics of the webpage categorization dataset

	Number of instances	Feature vector dimension	Number of labels	Label cardinality	Label density	Distinct label sets
Arts&Humanities	5000	462	26	1.636	0.063	462
Business&Economy	5000	438	30	1.588	0.053	161
Computers&Internet	5000	681	33	1.508	0.046	253
Education	5000	550	33	1.461	0.044	308
Entertainment	5000	640	21	1.420	0.068	232
Health	5000	612	32	1.662	0.052	257
Recreation&Sports	5000	606	22	1.423	0.065	322
Reference	5000	793	33	1.169	0.035	217
Science	5000	743	40	1.451	0.036	398
Social&Science	5000	1047	39	1.283	0.033	226
Society&Culture	5000	636	27	1.692	0.063	582

Table 5.1 summarizes the characteristics of the emotion, scene, yeast, medical and Enron datasets. We can remark that, for the medical and Enron datasets, the dimensions of feature vectors are very large as compared to the number of training instances. We applied the χ^2 statistic approach for feature selection on these two datasets, and we retained 20% of the most relevant features [122].

- The *webpage categorization dataset* has been investigated in [105][129]. The data were collected from the "yahoo.com" domain. Eleven different webpage categorization subproblems are considered, corresponding to 11 different categories: Arts and Humanities, Business and Economy, Computers and Internet, Education, Entertainment, Health, Recreation and Sports, Reference, Science, Social and Science, and Society and Culture. Each subproblem consists of 5000 documents. Over the 11 subproblems, the number of categories varies from 21 to 40 and the instance dimensionality varies from 438 to 1,047. Table 5.2 shows the statistics of the different subproblems within the webpage dataset.

5.4 Experiments on precise data

Before presenting the results, we explain hereafter the procedures used for parameter tuning and configuration of the proposed methods.

5.4.1 Parameter tuning

5.4.1.1 Parameter selection

DMLkNN, VERkNN and EMLkNN have one parameter in common that needs to be optimized: the number of neighbors k. In addition, DMLkNN has the fuzziness parameter δ, VERkNN and DMLkNN have the discounting parameter γ that also have to be fixed. We fixed these parameters using grid search and by focusing on the accuracy measure. k was varied from 1 to 30, δ from 0 to k, and γ from 0 to 3 with 0.05 step. $k = 10$ with $\delta = 2$ for DMLkNN and with $\gamma = 0.1$ for VERkNN and EMLkNN seem to be a good tuning for the proposed methods.

After five-fold cross-validation, Figure 5.1 shows the accuracy measure on the emotion and yeast dataset as a function of δ for $k = 10$. The maximum of the accuracy measure is achieved for $\delta = 2$ on the both datasets. As we can see, $\delta = 3$ can also be a candidate.

Figures 5.2 and 5.3 show the accuracy measure on the emotion dataset as a function of γ for $k = 10$, and as a function of k for $\gamma = 0.1$, using the VERkNN and EMLkNN methods, respectively. It is clear that $k = 10$ and $\gamma = 0.1$ is a good parametrization for the both methods. For VERkNN, the hybrid rule of combination was used, and for EMLkNN, we used the conjunctive rule. We will justify this choice in the next two sections.

5.4.1.2 Configuration of VERkNN

For VERkNN, two rules of combination can be used: the hybrid rule **and/or**, and the disjunctive rule **or**,. There also exist two approaches to generate verity and rebuff measures from precise labeled data: the direct and fuzzy approaches (see Section 3.4).

When using the fuzzy approach, we need to fix the number of neighbors k' to be taken into account in order to determine the appropriate labeling for each training instance. Figure 5.4 shows the *Accuracy* measure on the emotion dataset for different values of k'. We can remark that, for values larger than 5, parameter k' has no significant

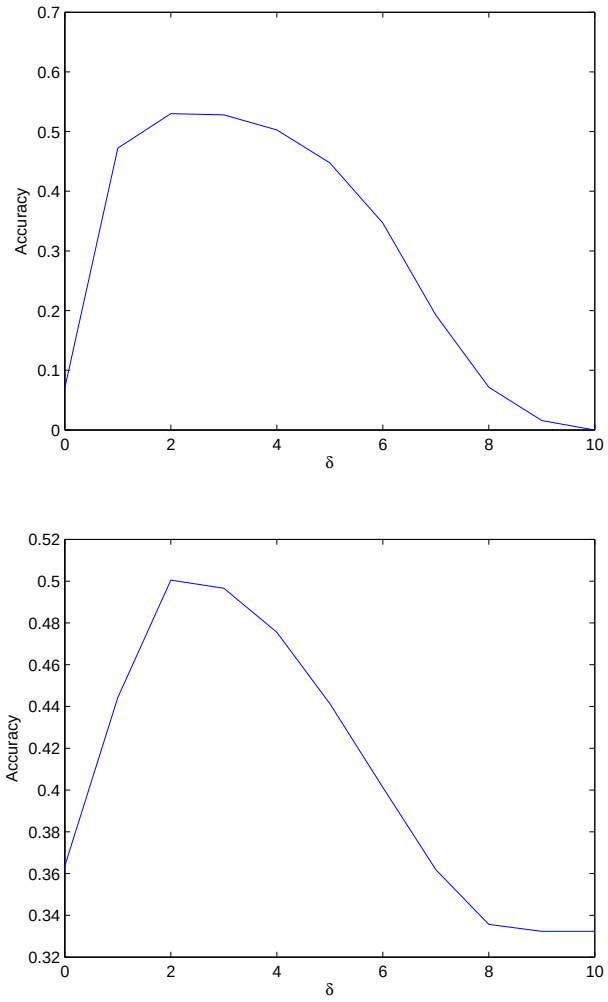

Figure 5.1: The accuracy measure of DMLkNN as a function of δ for $k = 10$, on the emotion dataset (top), and on the yeast dataset (bottom).

influence on the results when using the hybrid and disjunctive rules of combination. In the following, when using the fuzzy approach to determine the verity and rebuff distributions, the number of neighbors k' will be fixed to 8.

Tables 5.3 and 5.4 show a comparison on the emotion and yeast datasets respectively, between the hybrid and disjunctive rules of combination using the direct and fuzzy labeling approaches. The conclusion that can be drawn from these results is that the fuzzy labeling approach improves the performance of the VERkNN method, and for

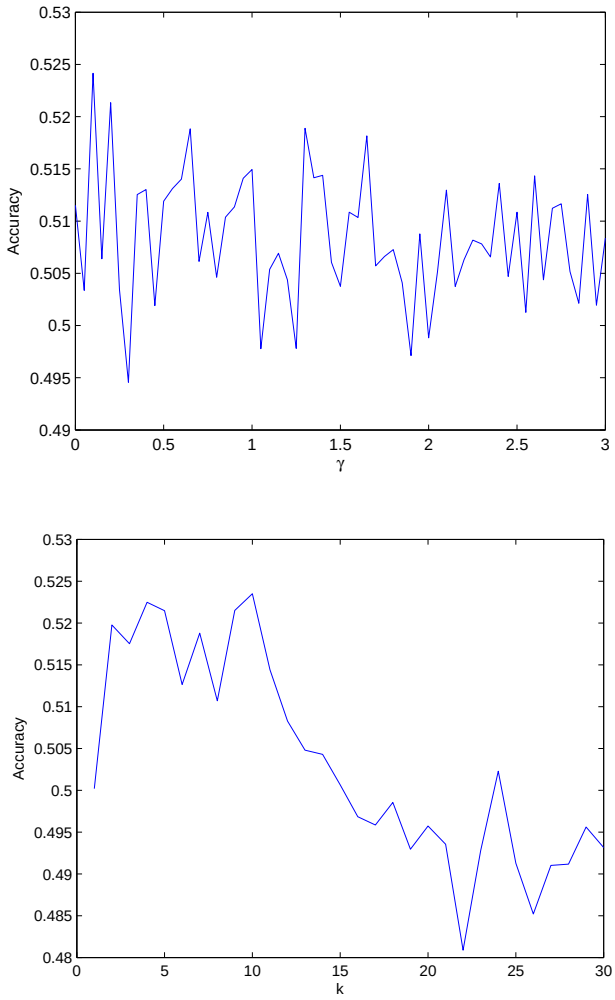

Figure 5.2: The accuracy measure of VERkNN on the emotion dataset as a function of γ for $k = 10$ (top), and as function of k for $\gamma = 0.1$ (bottom).

$k = 10$ and $\gamma = 0.1$, the hybrid rule of combination provide the best results on the two datasets.

5.4.1.3 Configuration of EMLkNN

For the EMLkNN method, a precisely labeled training instance $(\mathbf{x}_i, Y_i)$, is represented by $(\mathbf{x}_i, A_i, B_i)$, where $A_i = y_i$, and $B_i = \overline{Y}_i$ (see Section 4.6).

Tables 5.5 and 5.6 show a comparison on the emotion and yeast datasets respectively,

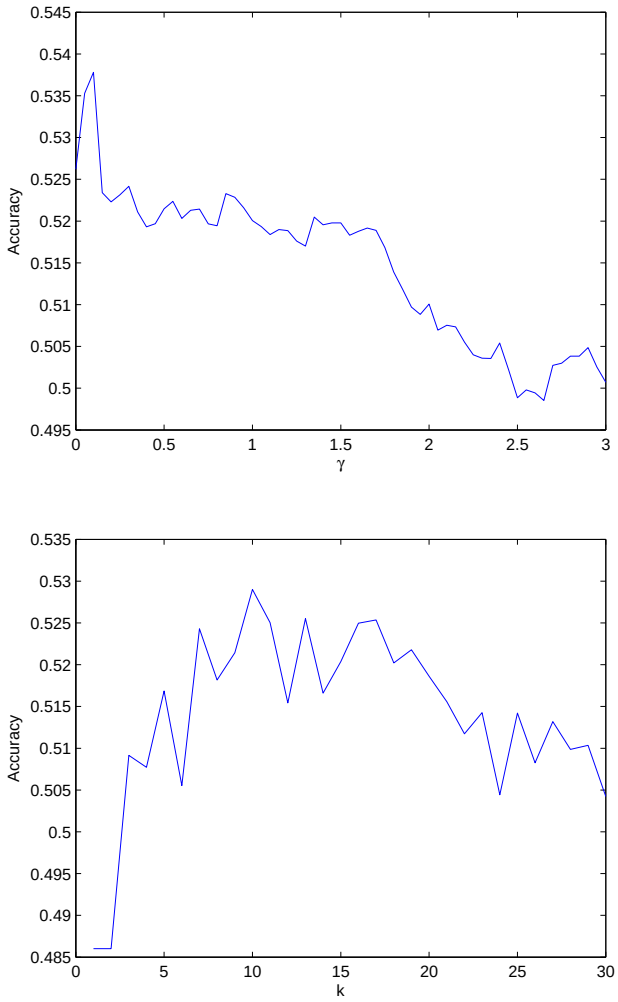

Figure 5.3: The accuracy measure of EMLkNN on the emotion dataset as a function of γ for $k = 10$ (top), and as function of k for $\gamma = 0.1$ (bottom).

between the consensus, conjunctive and cautious rules of combination. We can remark that, on both datasets, the results obtained by using the consensus and the conjunctive rules are very close, with a slight advantage to the conjunctive rule, and are better than the results obtained when using the cautious rule. In further experiments, the conjunctive rule of combination will be used because it is computationally faster than the consensus rule due to its associativity propriety.

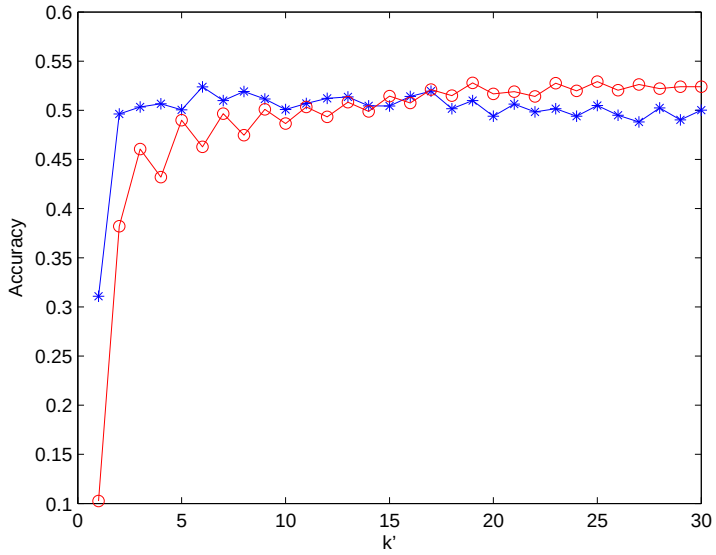

Figure 5.4: The accuracy measure on the emotion dataset for the VERkNN algorithm as a function of k', using the hybrid rule (*) and disjunctive rule (o) of combination.

Table 5.3: VERkNN on the emotion dataset

	VERkNN **and/or** (direct approach)	VERkNN **and/or** (fuzzy approach)	VERkNN **or** (direct approach)	VERkNN **or** (fuzzy approach)
Accuracy$^+$	0.511	**0.554**	0.429	0.503
Precision$^+$	0.605	**0.666**	0.433	0.539
Recall$^+$	0.677	0.626	**0.969**	0.855
F1$^+$	0.639	**0.645**	0.598	0.661
Hamming Loss$^-$	0.249	**0.222**	0.447	0.313
One-Error$^-$	0.382	**0.368**	0.645	0.549
Coverage$^-$	2.314	**2.296**	3.214	2.988
Ranking Loss$^-$	0.397	**0.388**	0.953	0.827
Average Precision$^+$	0.723	**0.745**	0.555	0.608

+(-): "the higher (smaller) the value the better the performance".

5.4.2 Results and discussion

We compared the proposed methods with three existing multi-label classification methods that were shown to have good performances and that were reported in Chapter 1: MLkNN [129] that is the closest to our methods, MLRBF [126] derived from radial basis function neural networks, and Rank-SVM [39] that is based on the traditional support vector machine. For each compared algorithm, the parameter tuning suggested

Table 5.4: VERkNN on the yeast dataset

	VERkNN **and/or** (direct approach)	VERkNN **and/or** (fuzzy approach)	VERkNN **or** (direct approach)	VERkNN **or** (fuzzy approach)
Accuracy$^+$	0.477	**0.512**	0.402	0.495
Precision$^+$	0.599	**0.665**	0.409	0.536
Recall$^+$	0.602	0.595	**0.955**	0.839
F1$^+$	0.601	**0.627**	0.573	0.654
Hamming Loss$^-$	0.244	**0.213**	0.472	0.300
One-Error$^-$	0.438	**0.285**	0.544	0.349
Coverage$^-$	6.844	**6.829**	10.318	10.212
Ranking Loss$^-$	**0.284**	0.285	0.885	0.721
Average Precision$^+$	0.705	**0.724**	0.476	0.546

+(-): the higher (smaller) the value, the better the performance.

Table 5.5: EMLkNN on emotion dataset

	EMLkNN (consensus rule)	EMLkNN (conjunctive rule)	EMLkNN (cautious rule)
Accuracy$^+$	0.544	**0.561**	0.470
Precision$^+$	**0.678**	0.676	0.645
Recall$^+$	0.626	**0.660**	0.532
F1$^+$	0.652	**0.667**	0.583
Hamming Loss$^-$	**0.194**	0.196	0.212
One-Error$^-$	**0.262**	0.267	0.278
Coverage$^-$	1.784	**1.783**	1.845
Ranking Loss$^-$	**0.165**	0.166	0.175
Average Precision$^+$	**0.804**	0.800	0.791

Table 5.6: EMLkNN on yeast dataset

	EMLkNN (consensus rule)	EMLkNN (conjunctive rule)	EMLkNN (cautious rule)
Accuracy$^+$	0.520	**0.525**	0.488
Precision$^+$	**0.692**	0.688	0.682
Recall$^+$	0.608	**0.613**	0.579
F1$^+$	0.647	**0.648**	0.623
Hamming Loss$^-$	**0.197**	0.198	0.206
One-Error$^-$	**0.236**	0.238	0.243
Coverage$^-$	6.522	**6.486**	6.614
Ranking Loss$^-$	0.186	**0.185**	0.189
Average Precision$^+$	0.758	**0.759**	0.751

in the literature were used: for ML-kNN, k was set to 10 [129]; for MLRBF, the fraction parameter was set to 0.01 and the scaling factor to 1 [126]; finally, a polynomial kernel was used for RankSVM [39].

For all k-NN based algorithms, the Euclidean distance was used. Laplace smoothing

was used for MLkNN and DMLkNN.

Five repetitions of ten-fold cross-validation were performed on each dataset. Tables 5.7 to 5.11 report the detailed results in terms of the different evaluation metrics for the emotion, scene, yeast, medical and Enron datasets, respectively. On the webpage dataset, ten-fold cross validation was performed on each subproblem, and Table 5.12 reports the average results.

For each method, the mean values of the different evaluation criteria as well as the standard deviations (std) are mentioned in the tables. A two-tailed paired t-test at 5% significance level was performed in order to determine the statistical significance of these results in comparison with the best performances indicated in bold. In addition, for each dataset, the methods were ranked in decreasing order of performance. The average ranks over the different evaluation criteria are reported in the tables.

On the emotion, scene and yeast datasets, the DMLkNN method had the best average performance, followed by EMLkNN and MLRBF. On the medical, Enron and webpage datasets, MLRBF performed better than the other methods, followed by DMLkNN and MLkNN.

From the presented experimental results, the following observations can be induced:

- VERkNN and EMLkNN perform better in terms of predicted-based metrics than in terms of ranking-based metrics. These methods work by combining information about the labeling of the nearest neighbors of each instance to classify, thus, they address mainly the pertinence of the predicted sets of labels instead of the ranking of all labels.

- DMLkNN performs better than MLkNN in terms of all ranking-based metrics and on all datasets. MLkNN gives better predicted-based measures on datasets from text categorization domain: the medical and Enron datasets.

- The proposed methods have a good performance and are more competitive with the other algorithms on datasets with high label density, such on the emotion and yeast datasets. In fact, the DMLkNN method takes into account label correlation. Moreover, the EMLkNN and VERkNN methods handling multi-labeled data directly, are intrinsically able to also capture relations between labels. Indeed, these methods will perform better on datasets with high label multiplicity, in which labels may be potentially more correlated.

Table 5.7: Experimental results (mean±std) on the emotion dataset

	DMLkNN	VERkNN	EMLkNN	MLkNN	MLRBF	RankSVM
Acc$^+$	0.562±0.029○	0.538±0.029●	**0.564±0.032**	0.536±0.032●	0.548±0.029●	0.403±0.027●
Prec$^+$	**0.691±0.032**	0.647±0.028●	0.676±0.034●	0.674±0.033●	0.686±0.037○	0.511±0.033●
Rec$^+$	0.653±0.030●	0.626±0.031●	**0.664±0.033**	0.622±0.041●	0.639±0.032●	0.538±0.032●
F1$^+$	**0.671±0.028**	0.645±0.027●	0.669±0.031○	0.648±0.033●	0.662±0.031●	0.524±0.029●
HLoss$^-$	**0.189±0.015**	0.222±0.014●	0.196±0.017●	0.197±0.015●	0.191±0.015○	0.288±0.016●
OErr$^-$	0.266±0.033●	0.368±0.043●	0.270±0.035●	0.285±0.035●	**0.255±0.045**	0.427±0.046●
Cov$^-$	**1.762±0.111**	2.281±0.147●	1.784±0.110●	1.803±0.115●	1.765±0.120○	2.425±0.129●
RLoss$^-$	0.161±0.019●	0.386±0.034●	0.168±0.021●	0.167±0.021●	**0.159±0.021**	0.278±0.020●
AvPrec$^+$	0.804±0.019○	0.745±0.027●	0.801±0.020○	0.794±0.022●	**0.809±0.024**	0.692±0.021●
Av Rank	1.5	4.6	2.5	3.8	2.1	5.8

+(-): the higher (smaller) the value, the better the performance.

●(○): statistically significant (non-significant) difference of performance as compared to the best result in bold, based on two-tailed paired t-test at 5% significance.

Table 5.8: Experimental results (mean±std) on the scene dataset

	DMLkNN	VERkNN	EMLkNN	MLkNN	MLRBF	RankSVM
Acc$^+$	0.676±0.015●	0.639±0.017●	**0.706±0.015**	0.668±0.020●	0.631±0.016●	0.436±0.015●
Prec$^+$	0.704±0.017●	0.659±0.019●	**0.735±0.016**	0.695±0.021●	0.652±0.017●	0.452±0.018●
Rec$^+$	0.677±0.015●	**0.772±0.017**	0.707±0.015●	0.687±0.024●	0.644±0.017●	0.661±0.017●
F1$^+$	0.690±0.016●	0.687±0.018●	**0.716±0.016**	0.683±0.023●	0.649±0.017●	0.508±0.017●
HLoss$^-$	**0.084±0.004**	0.120±0.004●	0.092±0.004●	0.087±0.003○	0.086±0.003○	0.163±0.004●
OErr$^-$	0.219±0.017●	0.319±0.016●	0.246±0.015●	0.228±0.016●	**0.206±0.015**	0.298±0.016●
Cov$^-$	0.461±0.035○	0.725±0.040●	0.527±0.030●	0.476±0.035●	**0.451±0.041**	1.187±0.043●
RLoss$^-$	**0.071±0.007**	0.160±0.009●	0.098±0.007●	0.077±0.009○	0.072±0.008○	0.120±0.010●
AvPrec$^+$	0.869±0.010○	0.804±0.010●	0.853±0.009●	0.865±0.009●	**0.876±0.009**	0.798±0.011●
Av Rank	2.1	4.3	2.7	3.1	2.8	5.7

+(-): the higher (smaller) the value, the better the performance.

●(○): statistically significant (non-significant) difference of performance as compared to the best result in bold, based on two-tailed paired t-test at 5% significance.

5.5 Experiments on imperfect data

Each of these datasets was constructed in such a way that each instance $\mathbf{x}_i$ is assigned a well-known set of labels Y_i. This choice may sometimes be questioned since in some cases, as with the emotion and scene datasets, there is no ground truth and the data have been labeled subjectively by one or several experts. In such a situation, uncertainty in class labels will inevitably exist due to conflicts between experts or lack of confidence that an expert may express. The veristic variable framework and the proposed evidential set-valued formalism allow us to represent and exploit expert knowledge. To assess the

Table 5.9: Experimental results (mean±std) on the yeast dataset

	DMLkNN	VERkNN	EMLkNN	MLkNN	MLRBF	RankSVM
Acc$^+$	0.511±0.011•	0.512±0.010•	**0.525±0.012**	0.508±0.014•	0.510±0.011•	0.474±0.019•
Prec$^+$	**0.726±0.014**	0.665±0.010•	0.692±0.013•	0.724±0.015•	0.703±0.013•	0.481±0.085•
Rec$^+$	0.577±0.012•	0.595±0.012•	**0.613±0.012**	0.578±0.017•	0.594±0.012•	0.541±0.066•
F1$^+$	0.613±0.011•	0.627±0.010•	**0.648±0.011**	0.612±0.014•	0.616±0.011•	0.502±0.052•
HLoss$^-$	**0.192±0.005**	0.229±0.005•	0.198±0.005•	0.194±0.005○	0.197±0.005•	0.204±0.010•
OErr$^-$	**0.226±0.021**	0.285±0.015•	0.238±0.016•	0.230±0.017○	0.239±0.019•	0.241±0.029•
Cov$^-$	**6.240±0.104**	6.829±0.132•	6.486±0.124•	6.275±0.100•	6.489±0.136•	7.027±0.489•
RLoss$^-$	**0.165±0.007**	0.284±0.011•	0.185±0.007•	0.167±0.006○	0.175±0.008•	0.189±0.015•
AvPrec$^+$	**0.770±0.010**	0.724±0.009•	0.759±0.008•	0.765±0.010•	0.758±0.011•	0.752±0.022•
Av Rank	2	4.3	2.6	2.9	3.4	5.6

+(-): the higher (smaller) the value, the better the performance.

•(○): statistically significant (non-significant) difference of performance as compared to the best result in bold, based on two-tailed paired t-test at 5% significance.

Table 5.10: Experimental results (mean±std) on the medical dataset

	DMLkNN	VERkNN	EMLkNN	MLkNN	MLRBF	RankSVM
Acc$^+$	0.548±0.031•	0.546±0.023•	0.628±0.035•	0.598±0.038•	**0.689±0.029**	0.462±0.042•
Prec$^+$	0.607±0.035•	0.596±0.025•	0.694±0.037•	0.657±0.041•	**0.713±0.031**	0.502±0.039•
Rec$^+$	0.558±0.030•	**0.723±0.029**	0.644±0.036•	0.623±0.038•	0.702±0.025○	0.549±0.037•
F1$^+$	0.571±0.032•	0.638±0.024•	0.656±0.036•	0.629±0.039•	**0.709±0.027**	0.520±0.037•
HLoss$^-$	0.015±0.001•	0.028±0.002•	0.017±0.002•	0.016±0.001•	**0.011±0.001**	0.204±0.004•
OErr$^-$	0.251±0.029•	0.474±0.037•	0.266±0.036•	0.252±0.026•	**0.141±0.024**	0.241±0.039•
Cov$^-$	2.664±0.447•	9.927±0.982•	3.261±0.451•	2.719±0.482•	**1.458±0.296**	4.027±0.786•
RLoss$^-$	0.040±0.009•	0.551±0.031•	0.101±0.016•	0.041±0.008•	**0.020±0.004**	0.189±0.021•
AvPrec$^+$	0.806±0.023•	0.522±0.029•	0.797±0.021•	0.802±0.019•	**0.896±0.014**	0.752±0.032•
Av Rank	3.1	4.8	3.2	3.1	1.1	5.5

+(-): the higher (smaller) the value, the better the performance.

•(○): statistically significant (non-significant) difference of performance as compared to the best result in bold, based on two-tailed paired t-test at 5% significance.

performances of the proposed methods in such situations, we randomly simulated an imperfect labeling process in order to generate imperfectly labeled data from precisely labeled ones.

5.5.1 Labeling process

Let $Y_i \subseteq \mathcal{Y}$ be the true label set of an instance $\mathbf{x}_i$, and let $\mathbf{y}_i = (y_{i1}, \ldots, y_{iQ})$ be the vector of $\{-1, 1\}^Q$ such that $y_{iq} = 1$ if $\omega_q \in Y_i$ and $y_{iq} = -1$ otherwise. For each instance $\mathbf{x}_i$ and each class ω_q, we generated a probability of error $p_{iq} = p'_{iq}/2$, where p'_{iq} was taken from a beta distribution with parameters $a = b = 0.5$ (this is a bimodal distribution with modes at 0 and 1), and we changed y_{iq} to $-y_{iq}$ with probability p_{iq},

Table 5.11: Experimental results (mean±std) on the Enron dataset

	DMLkNN	VERkNN	EMLkNN	MLkNN	MLRBF	RankSVM
Acc$^+$	0.341±0.037•	0.319±0.018•	0.361±0.016•	0.352±0.029•	**0.408±0.021**	0.269±0.044•
Prec$^+$	0.621±0.048•	0.529±0.030•	0.568±0.025•	0.615±0.030•	**0.643±0.025**	0.491±0.083•
Rec$^+$	0.349±0.037•	0.363±0.023•	0.401±0.019•	0.393±0.032•	**0.486±0.016**	0.341±0.061•
F1$^+$	0.427±0.037•	0.414±0.019•	0.444±0.018•	0.439±0.030•	**0.553±0.018**	0.398±0.064•
HLoss$^-$	0.052±0.001•	0.058±0.001•	0.055±0.002•	0.053±0.001•	**0.047±0.001**	0.085±0.012•
OErr$^-$	0.308±0.024•	0.438±0.053•	0.341±0.028•	0.310±0.029•	**0.278±0.018**	0.850±0.269•
Cov$^-$	**13.134±0.586**	29.265±1.011•	20.293±0.916•	13.199±0.588○	14.206±0.713•	26.804±2.712•
RLoss$^-$	**0.091±0.006**	0.486±0.027•	0.258±0.022•	0.092±0.006○	0.095±0.005○	0.273±0.070•
AvPrec$^+$	0.630±0.016•	0.365±0.024•	0.594±0.015•	0.629±0.017•	**0.688±0.014**	0.264±0.104•
Av Rank	2.5	4.8	3.3	2.7	1.4	6

+(-): the higher (smaller) the value, the better the performance.

•(○): statistically significant (non-significant) difference of performance as compared to the best result in bold, based on two-tailed paired t-test at 5% significance.

Table 5.12: Experimental results (mean±std) on the webpage dataset

	DMLkNN	VERkNN	EMLkNN	MLkNN	MLRBF	RankSVM
Acc$^+$	0.296±0.204•	0.323±0.168•	0.339±0.168○	0.285±0.184•	**0.398±0.146**	0.234±0.171•
Prec$^+$	0.351±0.257•	0.358±0.206•	0.399±0.193○	0.340±0.227•	**0.462±0.171**	0.228±0.212•
Rec$^+$	0.308±0.205•	0.315±0.173•	0.353±0.188○	0.291±0.189•	**0.407±0.153**	0.276±0.186•
F1$^+$	0.319±0.219•	0.322±0.181•	0.362±0.182○	0.304±0.198•	**0.421±0.156**	0.249±0.195•
HLoss$^-$	0.041±0.014•	0.054±0.022•	0.056±0.023•	0.043±0.015•	**0.039±0.013**	0.043±0.014•
OErr$^-$	0.466±0.165•	0.066±0.213•	0.797±0.262•	0.474±0.157•	**0.375±0.120**	0.440±0.143•
Cov$^-$	4.069±1.255○	9.021±3.662•	12.217±4.985•	4.097±1.237○	**4.689±1.403**	7.508±2.396•
RLoss$^-$	**0.099±0.046**	0.653±0.185•	0.761±0.196•	0.102±0.045○	0.107±0.039○	0.193±0.065•
AvPrec$^+$	0.630±0.120○	0.423±0.159•	0.358±0.162•	0.625±0.116○	**0.688±0.092**	0.601±0.117•
Av Rank	2.8	4.1	4.2	3.8	1.2	4.5

+(-): the higher (smaller) the value, the better the performance.

•(○): statistically significant (non-significant) difference of performance as compared to the best result in bold, based on two-tailed paired t-test at 5% significance.

resulting in a *noisy label vector* $\mathbf{y}'_i = (y'_{i1}, \ldots, y'_{iQ})$. Each number p_{iq} represents the probability that the membership of instance $\mathbf{x}_i$ to class ω_q has been wrongly assessed by the expert. This number may be turned into a degree of confidence c_{iq} by the transformation:

$$c_{iq} = 1 - 2p_{iq},$$

where $c_{iq} = 1$ means that the expert is totally sure about the membership ($y'_{iq} = 1$) or non membership ($y'_{iq} = -1$) of instance $\mathbf{x}_i$ to class ω_q, while $c_{iq} = 0$ means that he is totally undecided about this membership. We assume that these numbers can be provided by the expert, which allows us to label each instance $\mathbf{x}_i$ by a pair of sets (A_i, B_i), or by a pair of verity and rebuff distributions $(\mathrm{Ver}_i, \mathrm{Rebuff}_i)$, as explained

below.

Labeling $\mathbf{x}_i$ by (A_i, B_i) Using the degrees of confidence, we derive the imprecise label vector $\mathbf{y}''_i = (y''_{i1}, \ldots, y''_{iQ})$ from $\mathbf{y}'_i$ as follows:

$$y''_{iq} = \begin{cases} y'_{iq} & \text{if } c_{iq} \geq 0.6, \\ 0 & \text{otherwise.} \end{cases}$$

Such a vector of $\{-1, 0, 1\}^Q$ encodes an ordered pair (A_i, B_i) of disjoint subsets of $\mathcal{Y}$ such that:

$$\begin{cases} A_i = \{\omega_q \in \Omega \mid y''_{iq} = 1\}, \\ B_i = \{\omega_q \in \Omega \mid y''_{iq} = -1\}. \end{cases}$$

The set A_i then contains the classes ω_q that can be definitely assigned to the instance $\mathbf{x}_i$ with a high degree of confidence ($c_{iq} \geq 0.6$), while B_i is the set of classes which are definitely *not* assigned to $\mathbf{x}_i$. The remaining set $\mathcal{Y} \setminus (A_i \cup B_i)$ contains those classes about which the expert is undecided ($c_{iq} < 0.6$).

Labeling $\mathbf{x}_i$ by $(\mathbf{Ver}_i, \mathbf{Rebuff}_i)$ The verity and rebuff distributions of each instance $\mathbf{x}_i$ are generated as follows:

$$\mathrm{Ver}_i(\omega_q) = \begin{cases} c_{iq} & \text{if } y'_{iq} = 1 \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathrm{Rebuff}_i(\omega_q) = \begin{cases} c_{iq} & \text{if } y'_{iq} = -1 \\ 0 & \text{otherwise.} \end{cases}$$

$\mathrm{Ver}_i(\omega_q)$ represents the degree of confidence of the expert on assigning class ω_q to instance $\mathbf{x}_i$, while $\mathrm{Rebuff}_i(\omega_q)$ is the degree of confidence of the expert on *not* assigning ω_q to $\mathbf{x}_i$.

5.5.2 Results and discussions

As in the previous experiments, the proposed methods were compared to MLkNN, MLRBF and RankSVM. Each method was parameterized as in the case of precise data.

After simulated the labeling process as explained in Subsection 5.5.1, noisy data, where each instance $\mathbf{x}_i$ is labeled by $\mathbf{y}'_i$, and imprecise data, where each instance $\mathbf{x}_i$ is labeled by (A_i, B_i) for EMLkNN or by $(\mathrm{Ver}_i, \mathrm{Rebuff}_i)$ for VERkNN, were generated from each benchmark dataset.

Table 5.13 shows a comparative study between the different methods on noisy and imprecise data generated from the emotion dataset over ten trials. EMLkNN and VERkNN were applied on both noisy and imprecise data, while DMLkNN, MLkNN, MLRBF and RankSVM were only applied on the noisy data as it is not clear how imprecisely labeled data could be handled using these methods. It is clear that, in terms of the different evaluation metrics, VERkNN and EMLkNN perform better than the other methods with a significant advantage to the EMLkNN algorithm. We can also remark that the performances of EMLkNN and VERkNN were clearly improved when applied on the imprecise data instead of the noisy data. For example, in terms of the accuracy measure, the improvement was about 61% for EMLkNN, and about 23% for VERkNN. Similar results were obtained on the other datasets.

Figures 5.5 to 5.10 show the accuracy measure on the emotion, scene, yeast, medical, Enron and webpage datasets, respectively. For the webpage dataset, noisy and imprecise data were generated from each subproblem and the average performance out of the 11 different categorization problems was reported. For the other datasets, the performances for 10 different generations of noisy and imprecise data were mentioned. EML-kNN obviously dominates DMLkNN, MLkNN, MLRBF and RankSVM. VERkNN also performs better than the classical methods on the different datasets, but is always outperformed by EMLkNN.

These results demonstrate the ability of the proposed evidence formalism for set-valued variables to handle imprecisely labeled data in multi-label classification tasks. In fact, when the available learning data have been labeled subjectively by a pool of experts, noisy labels will be inevitably assigned to some instances due to conflicts or lack of knowledge. If an expert gives a degree of confidence about each assigned label, by using the EMLkNN method we are able to avoid risks of assigning wrongly some labels to an instance $\mathbf{x}_i$ when the degrees of confidence are not high. That explains the good performances of EMLkNN. In a similar manner, the VERkNN algorithm is also able to represent the knowledge given by the expert about the labeling of each instance in a proper manner close to the human language. This knowledge is represented by a verity distribution of positive information, and a rebuff distribution of negative information. However, the veristic formalism has not allowed us to reach the same level of performance as the set-valued evidence formalism.

Table 5.13: Experimental results on the *imperfectly* labeled emotion dataset

	Noisy data						Imprecise data	
	DMLkNN	VERkNN	EMLkNN	MLkNN	MLRBF	RankSVM	VERkNN	EMLkNN
Acc$^+$	0.282•	0.277•	0.271•	0.291•	0.271•	0.288•	0.339•	**0.438**
Prec$^+$	0.341•	0.331•	0.334•	0.342•	0.332•	0.333•	0.427•	**0.560**
Rec$^+$	0.523•	0.519•	0.524•	0.516•	0.515•	0.518•	0.558○	**0.575**
F1$^+$	0.424•	0.421•	0.408•	0.428•	0.404•	0.431•	0.484•	**0.567**
HLoss$^-$	0.503•	0.522•	0.498•	0.508•	0.501•	0.535•	0.402•	**0.284**
OErr$^-$	0.665•	0.705•	0.669•	0.692•	0.665•	0.677•	0.563•	**0.352**
Cov$^-$	3.429•	3.522•	3.479•	3.444•	3.488•	3.525•	3.089•	**2.479**
RLoss$^-$	0.481•	0.667•	0.489•	0.492•	0.493•	0.495•	0.472•	**0.314**
AvPrec$^+$	0.534•	0.529•	0.528•	0.523•	0.529•	0.518•	0.608•	**0.728**

+(-): the higher (smaller) the value, the better the performance.

•(○): statistically significant (non-significant) difference of performance as compared to the best result in bold, based on two-tailed paired t-test at 5% significance.

5.6 Conclusion

In this chapter, we have presented a comparison between the proposed multi-label classification methods and with some state-of-the-art methods. Different benchmark datasets and several evaluation criteria were used in the experiments. In our study, we investigated the cases of precise and impefect (noisy and imprecise) data.

In the case of precise data, the proposed methods are competitive with the other compared algorithms. We have focused out that the VERkNN and EMLkNN algorithms perform better in terms of predicted-based metrics than in terms of ranking-based metrics, and that they are more competitive on datasets with high label density. By taking into account the interdependencies between labels, the experiments demonstrate that DMLkNN improves the performance of the probabilistic k-NN rule for multi-label learning. DMLkNN performs better than MLkNN in terms of all ranking-based metrics and on all datasets.

In the case of noisy and imprecisely labeled data, VERkNN and EMLkNN perform significantly better than the other methods and on the different datasets, with a clear advantage to EMLkNN. These two methods seem to be able to handle practical situations where data have been labeled by experts, and allow us to model and exploit expert knowledge.

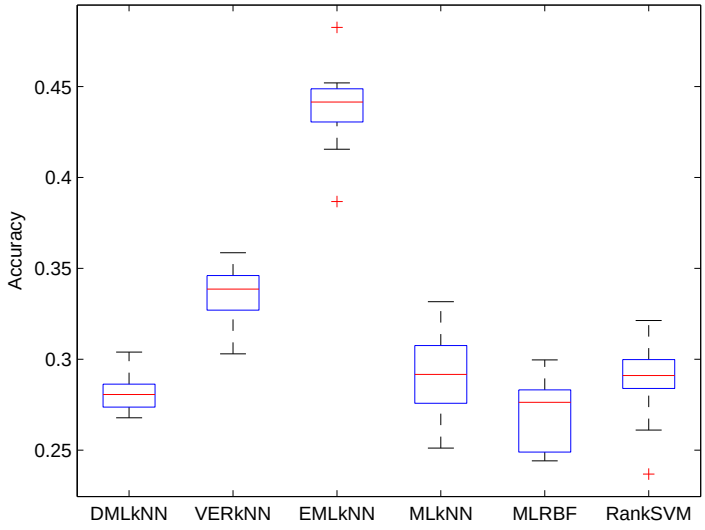

Figure 5.5: Box plots of the accuracy measure on the *imperfectly* labeled emotion dataset.

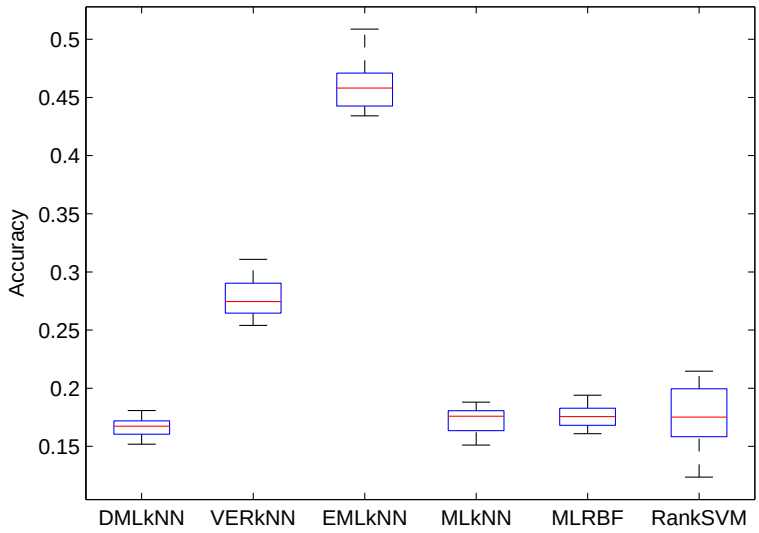

Figure 5.6: Box plots of the accuracy measure on the *imperfectly* labeled scene dataset.

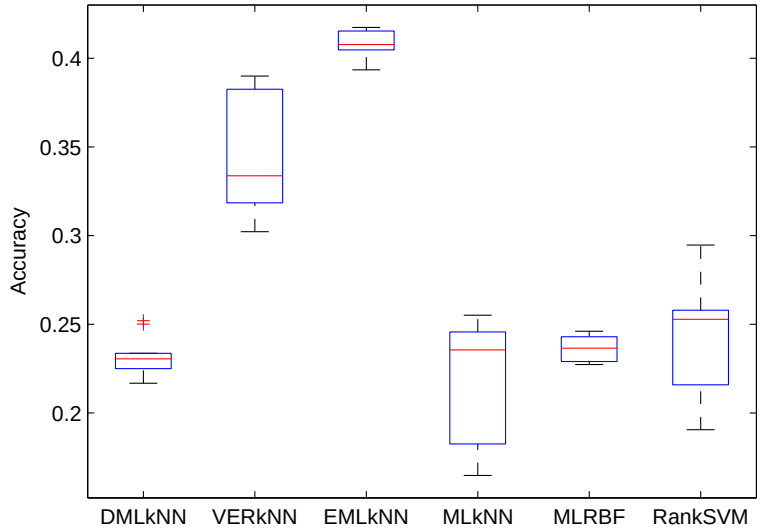

Figure 5.7: Box plots of the accuracy measure on the *imperfectly* labeled yeast dataset.

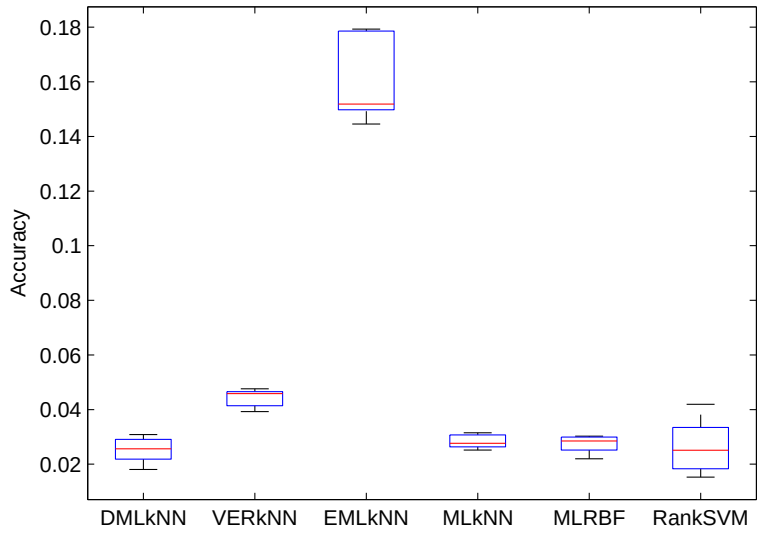

Figure 5.8: Box plots of the accuracy measure on the *imperfectly* labeled medical dataset.

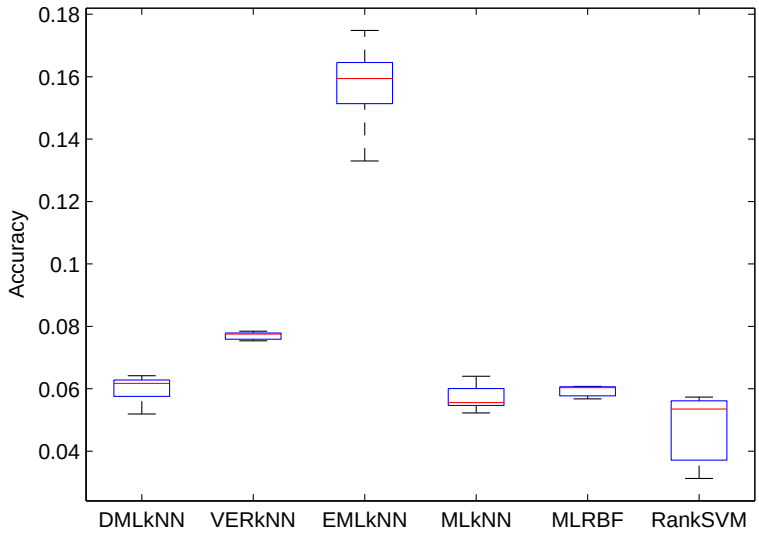

Figure 5.9: Box plots of the accuracy measure on the *imperfectly* labeled Enron dataset.

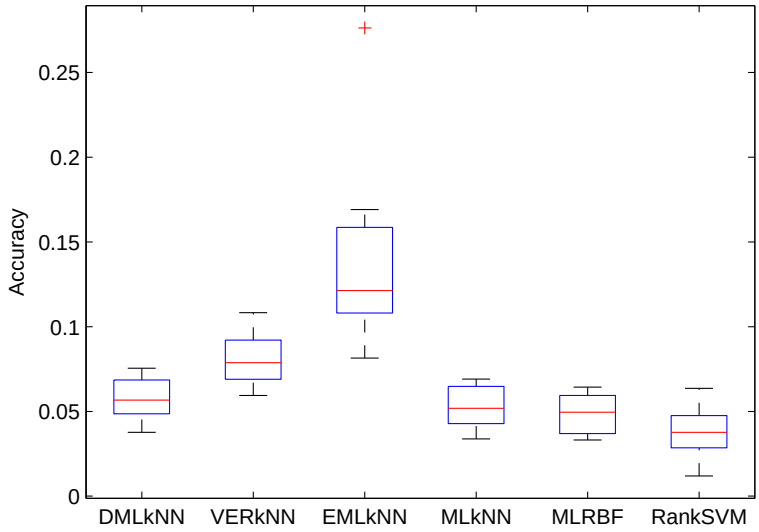

Figure 5.10: Box plots of the accuracy measure on the *imperfectly* labeled webpage dataset.

Conclusion and Perspectives

In this thesis, we addressed the problem of multi-label learning that become increasingly required by many real-world applications such as, text categorization, semantic scene analysis, bioinformatics, and music classification, where it is very frequent that instances belong to several classes at the same time. We have shown that there exist three main approaches for multi-label learning: Binary Relevance, Label Ranking, and Label Powerset. The basic idea of theses approaches consists in transforming a multi-label learning problem into one or more single-label learning problems. Taking label correlation into account has been shown to be a key challenge in multi-label learning, and it may improve the performance of multi-label classifiers.

A first method called DMLkNN has been introduced, based on a Bayesian learning. This method generalizes the state-of-the-art MLkNN algorithm by using a maximum a posteriori principle that models the relations between labels through statistical information extracted from the neighborhoods of instances to classify. An application on a simulated dataset asserted the ability of our method to capture correlation among labels, and experimental results on several benchmark datasets demonstrated the effectiveness of our method as compared to MLkNN and to other multi-label classifiers according to different evaluation metrics.

By the fact that class labels of multi-labeled data can be considered as veristic variables defined as fuzzy set-valued variables assuming multiple values simultaneously, the VERkNN method based on the theory of veristic variables has been proposed. In this method, the class label of each instance is represented by a verity distribution representing positive information about the membership of that instance to the different possible classes, and a rebuff distribution representing negative information.

The problem of multi-label learning has also been studied in the framework of the Dempster-Shafer theory. An evidence formalism to quantity uncertainty about set-

valued variables in general has been proposed, and has been applied in this thesis to build the EMLkNN method for multi-label classification. The basic idea of this formalism relies on the definition of the special lattice $(\mathcal{C}(\Omega), \subseteq)$ on which, most notions from Dempster-Shafer theory have been expressed with only a moderate increase of complexity as compared to the case of handling single-valued variables. This formalism has been shown to be more general than previous attempts to apply the Dempster-Shafer framework to represent uncertainty about set-valued variables. It has also been shown to be somewhat similar to, but arguably more general and flexible than other approaches introduced in the possibilistic framework; the veristic variable theory is one of them.

We have addressed the problem of learning from data with imprecise labels. Such problems occur in practical situations where data have been labeled by experts in the absence of ground truth. Due to conflits between experts and lack of confidence that they may expressed, it will be quite difficult to assign a precise set of labels to each instance. It has been shown that the VERkNN and EMLkNN methods allow us to model expert knowledge and represent imprecise labeling. The experimental results demonstrated the effectiveness of these methods in such kind of problems. VERkNN and EMLkNN perform significantly better than the other compared methods on all datasets, with a clear advantage to EMLkNN.

Perspectives

Some ideas proposed in this thesis may be improved and additional work in some research directions remains to be done. In the following paragraphs, we sketch a few of them.

As stated before, the experimental results proved that VERkNN and EMLkNN based on the veristic variable theory and the proposed set-valued evidence formalism respectively, are efficient methods to answer the problem of multi-label learning where data are labeled in an imprecise manner. It will be interesting to demonstrate theoretically the pertinence of these approaches, and to find real-world applications where applying these methods is specially adequate. Similarly, it remains to demonstrate mathematically the improvement in the performance of DMLkNN as compared to MLkNN in several datasets.

In this thesis, we addressed the problem of supervised multi-label learning with both precise and imprecise labeled data. This work may be pursued by investigating the problem of semi-supervised multi-label learning to manipulate both labeled and unlabeled instances at the same time. The problem of unsupervised multi-label learning seems also to be an important problem to resolve in order to handle totally unlabeled data including the special case where we have no prior knowledge about the target classes, i.e. multi-label clustering.

The k-nearest neighbor rule was used in this thesis to build multi-label classifiers based on the veristic variable and set-valued evidence frameworks. It will be interesting to develop other multi-label classification methods based on more sophisticated base classifiers, such as neural networks, support vector machines and linear discriminant analysis, conjunctively with these frameworks. In addition, we can also study the case of using an ensemble of multi-label classifiers and aggregating them in a probabilistic, possibilistic or evidential framework. It will be also interesting to study the problems of hierarchical and multi-instance multi-label learning under the frameworks cited above.

Finally, in Chapter 4, we have shown that most basic notions from the theory of belief functions can be defined on the special frame $\mathcal{C}(\Omega)$, such as plausibility and belief measures, canonical decomposition, combination of pieces of knowledge, etc. Other notions still not being investigated in order to show the possibility of extending them to the special frame, such as, conditioning and deconditioning, expressing partial knowledge on several set-valued variables taking values in different domains and generalizing the notions of marginalization and vacuous extension, studying informational comparisons of belief functions on set-valued variable as the plausibility ordering, specialization and generalization, etc [99]. It will be also interesting to find applications of the proposed set-valued evidence formalism other than multi-label classification. Querying databases and constructing a question-answering system may be a potential application [120].

Bibliography

[1] S. Aguzzoli, B. Gerla, and V. Marra. De Finetti's no-Dutch-book criterion for Gödel logic. *Studia Logica*, 90(1):25–41, 2008. 82

[2] R.T. Alves, M.R. Delgado, and A.A. Freitas. Multi-label hierarchical classification of protein functions with artificial immune systems. In *Proc. of the 3rd Brazilian Symposium on Bioinformatics*, LNBI 5167, pages 1–12, Santo Andre, Brazil, 2008. Springer. 11

[3] B. De Baetsa, E. Tsiporkovab, and R. Mesia. Conditioning in possibility theory with strict order norms. *Fussy Sets and Systems*, 106(2):221–229, 1999. 55

[4] Z. Barutcuoglu, R.E. Schapire, and O.G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006. 20

[5] S. Benferhat, D. Dubois, and H. Prade. From semantic to syntactic approaches to information combination in possibilistic logic. In B. Bouchon-Meunier, editor, *Aggregation and Fusion of Imperfect Information, Studies in Fuzziness and Soft Computing*, pages 141–151. Physica Verlag, 1997. 54

[6] C.M. Bishop. *Pattern recognition and Machine Learning*. Springer, New York, 2006. 7

[7] H. Blockeel, L. Schietgat, J. Struyf, S. Džeroski, and A. Clare. Decision trees for hierarchical multilabel classification: A case study in functional genomics. In *Proc. of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 18–29, Berlin, Germany, 2006. 20

[8] M.R. Boutell, J. Luo, X. Shen, and C.M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004. 8, 10, 13, 108

[9] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. 7

[10] Y. Chen and J.Z. Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939, 2004. 10

[11] W. Cheng and E. Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225, 2009. 7

[12] R.L.O. Cignoli, I.M.L. D'Ottaviano, and D. Mundici. *Algebraic foundations of many-valued reasoning*. Kluwer Academic, Dordrecht, 2000. 82

[13] A. Clare and R. D. King. Knowledge discovery in multi-label phenotype data. In *Proc. of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2001)*, LNCS 2168, pages 42–53, Freiburg, Germany, 2001. Springer. 11, 20

[14] B.R. Cobb and P.P. Shenoy. On the plausibility transformation method for translating belief function models to probability models. *International Journal of Approximate Reasoning*, 41(3):314–330, 2006. 75

[15] D.A. Cohn, Z. Ghahramani, and M.I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996. 13

[16] F.D. Comité, R. Gilleron, and M. Tommasi. Learning multi-label alternating decision tree from texts and data. In *Proc. of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM'03)*, LNCS 2734, pages 35–49, Leipzig, Germany, 2003. Springer. 9

[17] T.M. Cover and P.E. Hart. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. 29, 30

[18] K. Crammer and Y. Singer. A new family of online algorithms for category ranking. *Journal of Machine Learning Research*, 3:1025–1058, 2003. 17

[19] B.V. Dasarathy. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, CA, 1991. 29

[20] T. Denœux. A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics*, 25(5):804–813, 1995. 66, 67, 99, 100

[21] T. Denœux. Conjunctive and disjunctive combination of belief functions induced by non distinct bodies of evidence. *Artificial Intelligence*, 172:234–264, 2008. 71, 76, 77

[22] 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 B*, 36(6):1395–1406, 2006. 100

[23] T. Denœux and L.M. Zouhal. Handling possibilistic labels in pattern classification using evidential reasoning. *Fuzzy Sets and Systems*, 122(3):47–62, 2001. 100

[24] T.G. Dietterich, R.H. Lathrop, and T.L. Perez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997. 20, 21

[25] A. Dimou, G. Tsoumakas, V. Mezaris, I. Kompatsiaris, and I. Vlahavas. An Empirical Study Of Multi-Label Learning Methods For Video Annotation. In *Proc. of the 7th International Workshop on Content-Based Multimedia Indexing*, pages 19–24, Chania, Greece, 2009. 11

[26] C. Domeniconi, J. Peng, and D. Gunopulos. Locally adpative metric nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1281–1285, 2002. 29

[27] A. Dubois and H. Prade. On incomplete conjunctive information. *Computers and Mathematics with Applications*, 15(10):797–810, 1988. 97, 98

[28] A. Dubois and H. Prade. Representation and combination of uncertainty with belief functions and possibility measures. *Computational Intelligence*, 4:244–264, 1988. 74

[29] D. Dubois. Possibility theory and statistical reasoning. *Computational Statistics & Data Analysis*, 51(1):46–69, 2006. 53

[30] D. Dubois, L. Foulloy, G. Mauris, and H. Prade. Probability-possibility transformations, triangular fuzzy sets, and probabilistic inequalities. *Reliable Computing*, 10:273–297, 2004. 65

[31] D. Dubois and H. Prade. *Fuzzy Sets and Systems : Theory and Applications*. Academic Press, New York, 1980. 50

[32] D. Dubois and H. Prade. A set-theoretic view of belief functions: logical operations and approximations by fuzzy sets. *International Journal of General Systems*, 12:193–226, 1986. 49, 61, 94

[33] D. Dubois and H. Prade. *Possibility Theory*. Plenum Press, New York, 1988. 23, 53, 55

[34] D. Dubois and H. Prade. Fuzzy sets in approximate reasoning, part 1: Inference with possibility distributions. *Fuzzy Sets and Systems*, 40:143–202, 1991. 55, 56

[35] D. Dubois and H. Prade. Possibility theory: qualitative and quantitative aspects. In P. Smets, editor, *Handbook on Defeasible Reasoning and Uncertainty Management Systems, Vol. 1 : QuantiFed Representation of Uncertainty and Imprecision*, pages 169–226. Kluwer Academic, 1998. 52

[36] D. Dubois, H. Prade, and S.A. Sandri. On possibility/probability transformations. In R. Lowen and M. Roubens, editor, *Fuzzy Logic*, pages 103–112. Kluwer Academic, 1993. 55, 65

[37] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1973. 29

[38] S.A. Dudani. The distance-weighted k-nearest neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(4):325–327, 1976. 29

[39] A. Elisseeff and J. Weston. Kernel methods for multi-labelled classification and categorical regression problems. In *Advances in Neural Information Processing Systems*, volume 14, pages 681–687. MIT Press, 2002. 17, 106, 107, 108, 114, 115

[40] R.E. Fan and C.J. Lin. A study on threshold selection for multi-label classification. Technical report, National Taiwan University, 2007. 16

[41] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. 17

[42] K. Fukunaga. *Introduction to statistical pattern recognition.* Academic Press, USA, 2nd edition, 1990. 7, 27

[43] J. Fürnkranz, E. Hüllermeier, E. Loza Mencia, and K. Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008. 16, 17

[44] S. Gao, W. Wu, C.H. Lee, and T.S. Chua. A MFoM Learning Approach to Robust Multiclass Multi-Label Text Categorization. In *Proc. of 21st international conference on Machine learning (ICML'04)*, Banff, Canada, 2005. ACM. 9

[45] N. Ghamrawi and A. McCallum. Collective Multi-Label Classification. In *Proc. of the 14th ACM international conference on Information and knowledge management*, pages 198–200, Bremen, Germany, 2005. ACM. 9

[46] A.K. Ghosh. On nearest neighbor classification using adaptive choice of k. *Journal of Computational and Graphical Statistics*, 16(2):482–502, 2007. 30

[47] S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In *Proc. of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'04)*, LNCS 3056, pages 22–30, Sydney, Australia, 2004. Springer. 13, 14

[48] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*, volume 17, pages 513–520. MIT Press, 2005. 29

[49] M. Grabisch. Belief functions on lattices. *International Journal of Intelligent Systems*, 24:76–95, 2009. 24, 71, 78, 79, 80, 81

[50] M. Grabisch and C. Labreuche. Bi-capacities-I: definition, Möbius transform and interaction. *Fuzzy Sets and Systems*, 151:211–236, 2005. 85, 87, 90

[51] R. Hanson, J. Stutz, and P. Cheeseman. Bayesian classification theory. Technical Report FIA-90-12-7-01, NASA Ames Research Center, 1990. 27

[52] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor rule for classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):607–616, 1996. 29

[53] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897–1916, 2008. 16

[54] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999. 7

[55] S. Ji, L. Tang, S. Yu, and J. Ye. Extracting shared subspace for multi-label classification. In *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 381–389, Las Vegas, USA, 2008. ACM. 15

[56] R. Jin and Z. Ghahramani. Learning with multiple labels. In *Advances in Neural Information Processing Systems*, volume 15, pages 879–904. MIT Press, 2003. 20

[57] T. Joachims. Text categorization with support vector machines: learning with many relevant features. LNCS 1398, pages 137–142, Chemnitz, Germany, 1998. Springer. 13

[58] P.N. Juslin and J.A. Sloboda. *Music and Emotion: Theory and Research*. Oxford University Press, Oxford, UK, 2001. 10

[59] A karalič and V. Pirnat. Significance level based multiple tree classification. *Informatica*, 15(1):54–58, 1991. 12

[60] J.M. Keller, M.R. Gray, and J.A. Givens. A fuzzy k-nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(4):580–585, 1985. 62

[61] G.J. Klir and B. Parviz. Probability-Possibility Transformations: A Comparison. *International Journal of General Systems*, 21:291–310, 1992. 65

[62] S.B. Kotsiantis. Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, 31:249–268, 2007. 7

[63] T. Kroupa. Conditional probability on MV-algebras. *Fuzzy Sets and Systems*, 149(2):369–381, 2005. 82

[64] T. Kroupa. Representation and extension of states on MV-algebras. *Archive for Mathematical Logic*, 45(4):381–392, 2006. 82

[65] T. Kroupa. Geometry of uncertainty measures on formulas in Lukasiewicz logic. In *Algebra and Probability in Many-Valued Logics*, Darmstadt, Germany, Technische Universität Darmstadt, 2009. 82

[66] C. Labreuche and M. Grabisch. Modeling positive and negative pieces of evidence in uncertainty. In T.D. Nielsen and N.L. Zhang, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty (Proceedings of ECSQARU'03)*, pages 279–290, Aalborg, Denmark, 2003. Springer. 90

[67] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Advances in Neural Information Processing Systems*, volume 16, pages 553–560. MIT Press, 2004. 10

[68] D.D. Lewis, Y. Yang, T.G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004. 9

[69] T. Li and M. Ogihara. Toward intelligent music information retrieval. *IEEE Transactions on Multimedia*, 8(3):564–574, 2006. 10

[70] X. Li, L. Wang, and E. Sung. Multi-label SVM active learning for image classification. In *Proc. of the 2004 International Conference on Image Processing*, pages 2207–2210, Singapore, 2004. IEEE. 13

[71] L. Lu, D. Liu, and H.J. Zhang. Automatic mood detection and tracking of music audio signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):5–18, 2006. 10

[72] O. Maron and A.L. Ratan. Multiple-instance learning for natural scene classification. In *Proc. of the 15th International Conference on Machine Learning (ICML'98)*, pages 341–349, Madison, USA, 1998. 20

[73] G. Matheron. *Random Sets and Integral Geometry*. Wiley, New York, 1975. 95

[74] A. McCallum. Multi-Label Text Classification with a Mixture Model Trained by EM. In *Working Notes of the AAAI'99 Workshop on Text Learning*, pages 681–687, 1999. 7, 19

[75] A. McCallum, R. Rosenfeld, T.M. Mitchell, and A.Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proc. of the 5th International Conference on Machine Learning*, pages 359–367, San Francisco, USA, 1998. 9

[76] E. Loza Mencía and J. Fürnkranz. Pairwise learning of multilabel classifications with perceptronss. In *Proc. of the 2008 International Joint Conference on Neural Networks*, pages 2900–2907, Hong Kong, 2008. IEEE. 17

[77] T.M. Mitchell. *Machine Learning*. McGraw Hill, 1997. 7

[78] B. Monjardet. The presence of lattice theory in discrete problems of mathematical social sciences. Why? *Mathematical Social Sciences*, 46(2):103–144, 2003. 78, 79

[79] E. Moxley, T. Mei, and B.S. Manjunath. Video annotation through search and graph reinforcement mining. *IEEE Transactions on Multimedia*, 12(3):184–193, 2010. 11

[80] D. Mundici. Averaging the truth-value in Lukasiewicz logic. *Studia Logica*, 55(1):113–127, 1995. 82

[81] H.T. Nguyen. On random sets and belief functions. *Journal of Mathematical Analysis and Applications*, 65:531–542, 1978. 95

[82] H.T. Nguyen. *An Introduction to Random Sets*. Chapman and Hall/CRC Press, Boca Raton, Florida, 2006. 95

[83] K. Nigam, A. McCallum, S. Thrun, and T.M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine learning*, 39(2-3):103–134, 2000. 12

[84] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. 10

[85] J.S. Olsson. An analysis of the coupling between training set and neighborhood sizes for the k-NN classifier. In *Proc. of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 685–686, Seattle, USA, 2006. ACM. 30

[86] F. Peng, D. Schuurmans, and S. Wang. Augmenting Naive Bayes Classifiers with Statistical Language Models. *Information Retrieval*, 7(3-4):317–345, 2004. 40

[87] G.J. Qi, X.S. Hua, Y. Rui, J. Tang, T. Mei, and H.J. Zhang. Correlative multi-label video annotation. In *Proc. of the 15th ACM International Conference on Multimedia*, pages 17–26, Augsburg, Germany, 2007. ACM. 11

[88] J. Read, B. Pfahringer, and G. Holmes. Multi-label Classification using Ensembles of Pruned Sets. In *Proc. of the 8th IEEE International Conference on Data Mining (ICDM'08)*, pages 995–1000, Pisa, Italy, 2008. IEEE. 19

[89] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier Chains for Multi-label Classification. In *Proc. of the 20th European Conference on Machine Learning*, LNCS 5782, pages 254–269, Bled, Slovenia, 2009. Springer. 14, 108

[90] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, 7:1601–1626, 2006. 20

[91] R.E. Schapire and Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39:135–168, 2000. 7, 9, 16, 104

[92] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002. 9

[93] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, Princeton, N.J., 1976. 3, 23, 24, 59, 71, 72, 73, 74, 75, 89

[94] C. Shen, J. Jiao, B. Wang, and Y. Yang. Multi-Instance Multi-Label Learning For Automatic Tag Recommendation. In *Proc. of the 2009 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2009)*, pages 4910–4914, San Antonio, USA, 2009. 21

[95] P. Smets. The combination of evidence in the Transferable Belief Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):447–458, 1990. 61

[96] P. Smets. The Transferable Belief Model and random sets. *International Journal of Intelligent Systems*, 7:37–46, 1992. 95

[97] P. Smets. Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem. *International Journal of Approximate Reasoning*, 9:1–35, 1993. 74

[98] P. Smets. The canonical decomposition of a weighted belief. In *International Joint Conference on Artificial Intelligence*, pages 1896–1901, San Mateo, California, 1995. 75, 76

[99] P. Smets and R. Kenness. The Transferable Belief Model. *Artificial Intelligence*, 66:191–243, 1994. 23, 71, 72, 75, 129

[100] M.A. Tahir, J. Kittler, K. Mikolajczyk, and F. Yan. Improving Multilabel Classification Performance by Using Ensemble of Multi-label Classifiers. In *Proc. of the 9th International Workshop on Multiple Classifier Systems*, LNCS 5997, pages 11–21, Cairo, Egypt, 2010. Springer. 16

[101] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas. Multi-label classification of music into emotions. In *Proc. of the 9th International Conference on Music Information Retrieval*, Philadephia, PA, USA, 2008. 10, 107

[102] G. Tsoumakas and I. Katakis. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007. 11, 104, 105, 107

[103] G. Tsoumakas, I. Katakis, and I. Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. of the ECML/PKDD Workshop on Mining Multidimensional Data*, Antwerp, Belgium, 2008. 20

[104] G. Tsoumakas and I. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *Proc. of the 18th European Conference on Machine Learning*, pages 406–417, Warsaw, Poland, 2007. 18

[105] N. Ueda and K. Saito. Parametric mixture models for multi-label text. *Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA*, 15:721–728, 2003. 20, 109

[106] A. Vailaya, M. Figueiredo, A. Jain, and H.J. Zhang. Image classification for content-based indexing. *IEEE Transactions on Image Processing*, 10(1):117–130, 2001. 10

[107] V.N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998. 13

[108] C. Vens, J. Struyf, L. Schietgat, S. Dzeroski, and H. Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73:185–214, 2008. 20

[109] J. Wang, P. Neskovic, and L.N. Cooper. Neighborhood size selection in the k-nearest neighbor rule using statistical confidence. *Pattern Recognition*, 39:417–423, 2006. 30

[110] 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. 30

[111] Z. Wang, Y. Hu, and L.T. Chia. Multi-label learning by Image-to-Class distance for scene classification and image annotation. In *Proc. of the 9th ACM International Conference on Image and Video Retrieval (CIVR 2010)*, pages 105–112, Xi'an, China, 2010. ACM. 10

[112] K.Q. Weinberger, J. Blitzer, and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems*, volume 18, pages 1473–1480. MIT Press, 2006. 30

[113] D. Wettschereck and T.G. Dietterich. Locally Adaptive Nearest Neighbor Algorithm. In *Advances in Neural Information Systems*, volume 6, pages 184–191. MIT Press, 1994. 30

[114] A. Wieczorkowska, P. Synak, and Z.W. Ras. Multi-label classification of emotions in music. In *Proc. of the 2006 International Conference on Intelligent Information Processing and Web Mining (IIPWM'06)*, pages 307–315, Ustron, Poland, 2006. 10

[115] R.R. Yager. On different classes of linguistic variables defined via fuzzy subsets. *Kybernetes*, 13:103–110, 1984. 49, 94

[116] R.R. Yager. Set-based representations of conjunctive and disjunctive knowledge. *Information Sciences*, 41:1–22, 1987. 94

[117] R.R. Yager. Reasoning with conjunctive knowledge. *Fuzzy Sets and Systems*, 28:69–83, 1988. 95

[118] R.R. Yager. On the specificity of a possibility distribution. *Fuzzy Sets and Systems*, 50:279–292, 1992. 53

[119] R.R. Yager. Veristic variables. *IEEE Transactions on Systems, Man, and Cybernetics*, 30(1):71–84, 2000. 2, 23, 49, 56, 57, 95

[120] R.R. Yager. Querying databases containing multivalued attributes using veristic variables. *Fuzzy sets and systems*, 129(2):163–185, 2002. 58, 129

[121] Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1-2):69–90, 1999. 12, 105

[122] Y. Yang and J.O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In *Proc. of the 4th International Conference on Machine Learning*, pages 412–420, Nashville, TN, USA, 1997. Morgan Kaufmann. 109

[123] L.A. Zadeh. Fuzzy sets. *Information and control*, 8:338–353, 1965. 50

[124] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1:3–28, 1978. 23, 52, 53

[125] L.A. Zadeh. Toward a Theory of fuzzy information granulation and its certainty in human reasoning and fuzzy logic. *Fuzzy sets and Systems*, 90:111–127, 1997. 49

[126] M.L. Zhang. ML-RBF: RBF neural networks for multi-label learning. *Neural Processing Letters*, 29(2):61–74, 2009. 18, 114, 115

[127] M.L. Zhang and Z.J. Wang. MIMLRBF: RBF neural networks for multi-instance multi-label learning. *Neurocomputing*, 72(16-18):3951–3956, 2009. 21

[128] M.L. Zhang and Z.H. Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006. 17

[129] M.L. Zhang and Z.H. Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–3048, 2007. 13, 27, 34, 36, 44, 106, 109, 114, 115

[130] Z.H. Zhou. Multi-Instance Learning: A Survey. Technical report, National Laboratory for Novel Software Technology, Nanjing University, China, 2004. 21

[131] S. Zhu, X. Ji, W. Xu, and Y. Gong. Multi-labelled classification using maximum entropy method. In *Proc. of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 274–281, New York, 2005. ACM. 20

[132] L.M. Zouhal and T. Denoeux. An evidence-theoretic k-NN rule with parameter optimization. *IEEE Transactions on Systems, Man and Cybernetics C*, 28(2):263–271, 1998. 99, 100

[133] W. Zuo, D. Zhang, and K. Wang. On kernel difference-weighted k-nearest neighbor classification. *Pattern Analysis & Applications*, 11(3-4):247–257, 2008. 62