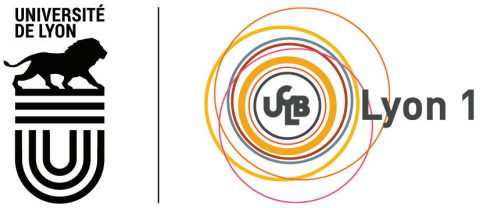
Ensemble multi-label learning in supervised and semi-supervised settings

Ouadie Gharroudi



º d'ordre NNT : 2017*LYSE*1333

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON

opérée au sein de

l'Université Claude Bernard Lyon 1

École Doctorale ED512

Infomath

Spécialité de doctorat : Informatique

Soutenue publiquement le 21/12/2017, par :

Ouadie GHARROUDI

Ensemble multi-label learning in supervised and semi-supervised settings

---

Devant le jury composé de :

Mustapha LEBBAH, Maître de Conférences HDR, Université Paris 13	Rapporteur
Pascale KUNTZ, Professeure, Université de Nantes	Rapporteure
Elisa FROMONT, Professeure, Université de Rennes 1	Examinatrice
Alexandre AUSSEM, Professeur, Université Lyon 1	Directeur de thèse
Haytham ELGHAZEL, Maître de Conférences, Université Lyon 1	Co-Directeur de thèse

Abstract

Multi-label learning is a specific supervised learning problem where each instance can be associated with multiple target labels simultaneously. Multi-label learning is ubiquitous in machine learning and arises naturally in many real-world applications such as document classification, automatic music tagging and image annotation.

In this thesis, we formulate the multi-label learning as an ensemble learning problem in order to provide satisfactory solutions for both the multi-label classification and the feature selection tasks, while being consistent with respect to any type of objective loss function.

We first discuss why the state-of-the art single multi-label algorithms using an effective committee of multi-label models suffer from certain practical drawbacks. We then propose a novel strategy to build and aggregate k-labelsets based committee in the context of ensemble multi-label classification. We then analyze the effect of the aggregation step within ensemble multi-label approaches in depth and investigate how this aggregation impacts the prediction performances with respect to the objective multi-label loss metric.

We then address the specific problem of identifying relevant subsets of features - among potentially irrelevant and redundant features - in the multi-label context based on the ensemble paradigm. Three wrapper multi-label feature selection methods based on the Random Forest paradigm are proposed. These methods differ in the way they consider label dependence within the feature selection process.

Finally, we extend the multi-label classification and feature selection problems to the semi-supervised setting and consider the situation where only few labelled instances are available. We propose a new semi-supervised multi-label feature selection approach based on the ensemble paradigm. The proposed model combines ideas from co-training and multi-label k-labelsets committee construction in tandem with an inner out-of-bag label feature importance evaluation.

Satisfactorily tested on several benchmark data, the approaches developed in this thesis show promise for a variety of applications in supervised and semi-supervised multi-label learning.

Contents

Chapter 1

Introduction

1.1 The scope of the thesis

Machine learning is a multidisciplinary field consisting of many contributing scientific domains related to computing, mainly Artificial Intelligence, Mathematics, Statistics, and Probability. In 1959, Arthur Samuel defined machine learning as "a field of study that gives computers the ability to learn without being explicitly programmed." During the last few decades, machine learning gained in popularity and become ubiquitous in various application domains such as recognition systems, natural language processing, and data mining [1]. With the broadening availability of large-scale data sets, machine learning is expected to play a significant role in everyday life by providing predictive solutions that generalize well from previously observed examples.

An important research field in machine learning is the task of inferring a function that can predict the best value for an output target variable given an input object (typically a vector of variables). This task is known as *Supervised learning*. The function is learned by exploring a set of observed examples (training examples) with an already identified input and output pairs. The idea is to take advantage of a limited number of observed examples to induce a mechanism that automatically annotates the output (the target variable) for a large set of examples or new unseen examples. In the traditional supervised learning context, there is only one target variable to predict. The supervised task is categorized as single-label classification when the target variable is discrete and categorized as regression when the target variable is continuous.

Multi-label classification has emerged as a natural extension to single-label classification in response to applications where examples are associated with multiple interdependent classes simultaneously. For example, a medical patient may be diagnosed with more than one health condition: 'asthma', 'diabetes', 'high blood pressure', and 'heart disease'. Likewise, an article can be categorized into multiple categories: 'education', 'business', 'technology', 'social', and

'science'. From a computational perspective, the multi-label classification aims to obtain a bi-partition ("on" and "off") of the set of all possible classes; the positive classes are referred to as labels, the so-called relevant labels of the instances. Under these circumstances, the one-label assignment assumption conducted by conventional single-label classification methods is not satisfied. First, each example can be associated with more than one label at the same time. Thus, the prediction model should correctly associate a collection of binary classifications to an unseen example. Second, the performance evaluation of the multi-label prediction are different; since that, a multi-label prediction could be partially correct (where some labels are correctly predicted), fully wrong (where all predictions are wrong), or fully correct (where all labels are correctly predicted).

Multi-label models also have to deal with other challenges such as the inherent labels dependencies, the computational complexity related of the model's inference, the large dimensions of the (input/output) spaces and the imbalance label representation where negative labels massively outnumber positive ones. Various multi-label algorithms have been developed in the literature [2] to cope these challenges. Tsoumakas and Katakis [3] summarized the multi-label classification algorithms into two categories depending on the manner in which they tackle the multi-label task, namely *problem transformation methods* [3, 4] and *algorithm adaptation methods* [5–8]. The first category transforms the multi-label learning task into either several binary classifications or one multi-class classification problem. *Algorithm adaptation methods*, on the other hand, extends specific learning models to handle the multi-labeled data.

Besides these two categories of multi-label algorithms a third category of meta-models distinguish itself as *ensemble multi-label models* [9]. Ensemble multi-label models are based on the top of a committee of single multi-label models with the goal of combining their outputs as a single prediction. This group of models aims to enhance the generalization ability of single-models by combining multiple ones to accomplish jointly one common task. The improvement of performances within this family of methods relies on the concept of diversity, stating that a good ensemble is a committee of models in which misclassified instances are different from one individual model to another. This paradigm has proved to be efficient in traditional single-label learning with a large body of work [10–14].

In the multi-label context, ensemble models have been suggested, not only to improve the predictive performance and the robustness of single-models, but also to overcome other issues that are specific to multi-labeled data (such as the learning complexity [15, 16], and the independence assumption over the target labels [17]). For example, to deal with a large number of labels while maintaining moderate learning complexity, Tsoumakas, and Katakis [15] proposed to construct a committee of multi-label models where each member is specialized in a subset of labels with the idea to combine their outputs in the prediction step.

In most of the studies in ensemble multi-label learning, the emphasis is generally on the way the committee is constructed, rather than on the combination step, and they often fail to provide the new ensemble model with the adequate multi-label combination strategy that is consistent with the committee construction. The combination is treated as in the traditional single-label ensemble models and often highlighted as a step that can only improve the predictions quality. In many works, a careful analysis of the combination step is lacking, thereby ignoring the peculiarity of the multi-label context, namely different committee structure and evaluation metrics. In fact, since multiple interdependent labels can be predicted simultaneously by each committee member, ensemble multi-label models cannot always rely on a straightforward combination scheme borrowed from the single-label learning.

1.2 Challenges and research goals

The main question studied in this dissertation is how to tackle multi-label learning problems through the ensemble paradigm. The thesis explores ensemble multi-label models construction including the diversity induction used to generate the base classifier committee and the aggregation of their predictions. The work also analyzes the type of loss metrics optimized by the state-of-the-art ensemble model and the influence of the different stages of the ensemble framework on their prediction quality. The thesis identifies some unique characteristics of the aggregation step and its connection with the loss function minimized by the ensemble model.

The main objective of this dissertation is to study in depth how ensemble approaches can be used effectively for multi-label learning and its related tasks, such as classification and feature selection in a supervised and a semi-supervised way. To accomplish this objective, this work is divided into two main parts.

The first focuses on ensemble multi-label classification problems, especially for the needs to optimize a particular loss metric. This raises a number of challenging questions:

- How to build a loss consistent ensemble multi-label model? Is it sufficient to consider the objective loss function exclusively in the committee construction?
- What is the role and the influence of the combination step in the ensemble of multi-label models? Should we combine base-classifier predictions with a specific combination strategy instead of a simple label-wise combination strategy?

These questions reflect the fundamental problem in multi-label classification that we therefore wish to address in this thesis. Our objective is to develop an efficient ensemble framework that

remains fair for all multi-label committee-based models while being consistent with a multi-label loss metric to minimize. We discuss different ensemble combination strategies addressing the loss consistency issues in the ensemble multi-label model and propose a new calibration algorithm adapting the ensemble prediction output to meet the objective loss function.

The second part aims to extend the ensemble multi-label framework to conduct multi-label feature selection, in the supervised and the semi-supervised way when only a few multi-label instances are available. This raises again the following questions:

- Can we efficiently use the power of ensemble methods to identify and remove the irrelevant features in a multi-label setting? Is there a link between the loss function minimized by the ensemble model and the model's feature importance estimation?
- Can we benefit from the ensemble paradigm advantages to tackle the multi-label feature selection in the semi-supervised context?

1.3 Contribution

The main novelty in this thesis is an efficient exploitation of the ensemble paradigm in the multi-label context. The thesis starts by addressing some shortcomings of the k-labelsets based ensemble multi-label approaches. We first propose a novel strategy to build and aggregate k-labelsets based committee in line with an objective multi-label loss function of interest.

Motivated by the results obtained in this part, we discuss in depth the effect of different aggregation strategies within various state-of-the-art ensemble multi-label approaches. Then, we investigate how these combinations strategies can effectively impact the performances of ensemble models especially when they are used in conjunction with a thresholding strategy that optimizes a multi-label performance measure of interest.

The second part of this thesis is dedicated to the problem of the multi-label feature selection based on the ensemble paradigm. We propose to evaluate the feature importance in multi-label data using three different wrapper approaches in a Random Forest style. These variants optimize different loss metrics depending on the way the label dependence is estimated. We also analyze how the optimized loss metrics (in the inner multi-label classifier) influences the relevance of a multi-label feature selection process.

Finally, the dissertation considers the problem of using a large amount of unlabeled data to improve the efficiency of feature selection in high dimensional multi-label data sets, when only a small set of labeled examples is available. We propose a new semi-supervised multi-label feature importance evaluation method, which combines ideas from co-training and random k-labelsets ensemble learning with a new permutation-based out-of-bag feature importance measure.

1.4 Thesis organization

This manuscript is intended to be self-contained. Readers familiar with machine learning concepts may skip Chapters 2, 3 and 6, which respectively present a comprehensive review of Multi-label Classification, ensemble learning and Multi-label feature selection approaches. Personal contributions are reported in Chapters 4, 5, 7 and 8.

In Chapter 2 we introduce the fundamentals of multi-label learning including both problem formulation and evaluation metrics. The chapter also reviews proposed multi-label classification approach with a scrutinized analysis over their optimized loss function.

In Chapter 3, we give an overview of ensemble learning with a focus on the state-of-the-art ensemble multi-label models. The goal is to provide the necessary background to understand the approaches presented in the latter parts of this thesis.

Chapter 4 and Chapter 5 present the main contributions of the thesis on multi-label classification. In particular, Chapter 4 presents our novel strategy to build and aggregate k-labelsets based committee in line with an objective multi-label loss function of interest.

Chapter 5 elaborates on the issue of base-classifier combination in various state-of-the-art ensemble multi-label approaches and discusses its impact on the performances of ensemble models especially when it is used in conjunction with a thresholding strategy that optimizes a multi-label performance measure of interest.

Chapter 6 reviews recent studies on supervised and semi-supervised multi-label feature selection.

Chapter 7 introduces the three Random Forest based multi-label feature selection methods and describes how variable importance used in Random Forest can be extended in multi-label context.

In Chapter 8, we propose a new proposed ensemble multi-label framework to help to solve the problem of multi-label feature selection in a semi-supervised multi-label way.

Chapter 9 concludes the thesis and outlines open research problems for further research directions.

Chapter 2

Multi-label learning

Multi-label learning is the extension of single-label classification in which the goal is to predict the set of relevant labels for a given input. This classification context is encountered in various fields, including text, multi-media, biology. It was introduced to cope complex learning problems of multi-class classification, with the aim to predict simultaneously a set of classes appointed as labels. The issue of learning from multi-label data has recently attracted significant attention from many researchers, and a considerable number of approaches have been proposed [2, 9, 18]. From a computational perspective, multi-label classification aims to obtain simultaneously a collection of binary classifications for each individual object. There are two broad categories of algorithms in multi-label learning, namely *a) problem transformation methods* and *b) algorithm adaptation methods*. The first category transforms the multi-label learning task into either several binary classifications or one multi-class classification problem. *Algorithm adaptation methods*, on the other hand, extend specific learning models to handle the multi-labeled data.

This chapter will be devoted to present the fundamental concept of the multi-label learning and to summarize the state-of-the-art of multi-label algorithms. The chapter also gives a first analysis about the loss metric optimized by several well-established multi-label models. It starts with a formal statement of the multi-label learning problem, then discusses the multi-label evaluation metrics and gives a survey of works related to the multi-label learning. Finally, the loss function optimized in the multi-label algorithms is discussed. The goal of the chapter is to provide the necessary background to understand the approaches presented in the upcoming parts of this thesis.

2.1 Multi-label terminology

In contrast to the traditional single-label learning, the target labels are not mutually exclusive in the multi-label context. Instances can be associated simultaneously with more than one label.

Let $\mathcal{X}$ denote the input (instance) space, and let $\mathcal{L} = \{\lambda_1, \lambda_2, ..., \lambda_q\}$ be a finite set of labels. Assuming that each training instance $\mathbf{x} \in \mathcal{X}$ is associated with a subset of labels l, where $l \subseteq \mathcal{L}$, this subset of labels is called *labelset* and denotes the relevant labels for $\mathbf{x}$. The remaining set of labels ($\mathcal{L} \setminus l$) represents, on the other hand, the set of irrelevant labels for $\mathbf{x}$. These sets of relevant are represented by a binary vector $\mathbf{y} = (y^1, y^2, \cdots, y^q)$, where $y^i = 1 \Leftrightarrow \lambda_i \in l$ and $y^i = 0\ otherwise$. The set of all possible subsets of labels (*i.e.* the powerset of $\mathcal{L}$: $\mathcal{P}(\mathcal{L})$) is denoted by $\mathcal{Y} = \{0, 1\}^q$ and represents the output (label) space.

Besides, instances in the input space can be described over a collection of f features which can be Boolean, discrete, or continuous or even a mixture thereof (*i.e.*, $\mathbf{x}_{(j)} = (x^1_{(j)}, \dots, x^M_{(j)}), \forall \mathbf{x}_{(j)} \in \mathcal{X}$, Where the bold is used to distinguish vectors from scalars). Thus, a multi-label sample is a join up of tuples from the descriptive space and the label space, $(\mathbf{x}_{(j)}, \mathbf{y}_{(j)}) \in \mathcal{X} \times \mathcal{Y}$. Table 2.1 shows the data set representation of a multi-label data set E consisting of n instances : $E = \{(\mathbf{x}_{(1)}, \mathbf{y}_{(1)}), \cdots, (\mathbf{x}_{(n)}, \mathbf{y}_{(n)})\}$.

	$\mathbf{X}^1$	$\mathbf{X}^2$	…	$\mathbf{X}^M$	$\mathbf{Y}^1$	$\mathbf{Y}^2$	…	$\mathbf{Y}^q$
$\mathbf{x}_{(1)}$	$x^1_{(1)}$	$x^2_{(1)}$	…	$x^M_{(1)}$	$y^1_{(1)}$	$y^2_{(1)}$	…	$y^q_{(1)}$
$\mathbf{x}_{(2)}$	$x^1_{(2)}$	$x^2_{(2)}$	…	$x^M_{(2)}$	$y^1_{(2)}$	$y^2_{(2)}$	…	$y^q_{(2)}$
⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋱	⋮
$\mathbf{x}_{(n)}$	$x^1_{(n)}$	$x^2_{(n)}$	…	$x^M_{(n)}$	$y^1_{(n)}$	$y^2_{(n)}$	…	$y^q_{(n)}$

TABLE 2.1: Multi-label data set

An important characteristic of a multi-label data set is the number of labels associated to each example. Depending on the application domain, this number of label can be large or small relatively to the number of all possible labels. Tsoumakas and Katakis [3] proposed two pertinent statistics that describes a multi-label data set: The *label cardinality* and the *label density*. The *label cardinality* indicates the average number of labels associated with each instance, while the *label density*, indicates the average proportion of labels associated with each example. Lets $|\mathbf{y}|$ denote the number of labels represented in $\mathbf{y}$. For a given data set $\mathcal{D}$ the statistics are defined as follows:

- The *label cardinality* (***Card***)

$$\boldsymbol{Card}(E) = \frac{1}{n}\sum_{j=1}^{n} |\mathbf{y}_{(j)}|$$

- The *label density* (**LD**)

$$\mathbf{LD}(E) = \frac{1}{n}\sum_{j=1}^{n} \frac{|\mathbf{y}_{(j)}|}{q}$$

Both statistics characterize the number of labels that describe the instances of a multi-label data set. The former is independent of the size of the label space, while the latter considers the number of labels q. Two multi-label data sets with exactly the same label cardinality and different label density might exhibit distinctive properties that impact the predictive multi-label model. The two statistics are related to each other : $\boldsymbol{Card}(E) = q \times \mathbf{LD}(E)$.

2.2 Multi-label learning: Formulation & Problem Statement

Assuming that $\mathbf{x}$ and $\mathbf{y}$ are jointly distributed according to some fixed but unknown probability distribution $P(\mathbf{x}, \mathbf{y})$ over $\mathcal{X} \times \mathcal{Y}$, the multi-label classification task is formulated as follows:

Given a training data, in the form of a finite set of paired observations $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$ generated by sampling according to the distribution $P(\mathbf{x}, \mathbf{y})$. The goal is to provide an estimator $\mathbf{h} : \mathcal{X} \to \mathcal{Y}$ which predicts the best value of an output $\mathbf{y}$ given an input $\mathbf{x}$. That is, the estimator $\mathbf{h}$ returns, for each $\mathbf{x} \in \mathcal{X}$, a predicted vector $\mathbf{h(x)} = (h^1(\mathbf{x}), h^2(\mathbf{x}), \cdots, h^q(\mathbf{x}))$ with the objective to generalize well beyond the training observations in the sense of minimizing the risk with respect to a specific loss metric. Basically, the learning model aims to minimize the expected risk of $\mathbf{h}$ with regard to some multi-label loss $L(\cdot)$, i.e.,

$$R_L(\mathbf{h}) = \mathbb{E}_{\mathbf{X},\mathbf{Y}}[L(\mathbf{Y}, \mathbf{h}(\mathbf{X}))] \tag{2.1}$$

In general, it is not easy to learn the $\mathbf{h}$ directly. In practice, one instead learns a real-valued vector function $\mathbf{s} : \mathcal{X} \to \mathbb{S}$, where the predicted score can be either $s(\mathbf{x}, \mathbf{y})$, so $\mathbb{S} = \mathbb{R}^{|\mathcal{Y}|}$; or $s(\mathbf{x}, \lambda_i)$, so $\mathbb{S} = \mathbb{R}^q$ and $\mathbf{s}(\mathbf{x}) = (s(\mathbf{x}, \lambda_1), \cdots, s(\mathbf{x}, \lambda_q))$. $s(\mathbf{x}, \mathbf{y})$ is the confidence of $\mathbf{y} \in \mathcal{Y}$, being the proper labelset of $\mathbf{x}$; and $s(\mathbf{x}, \lambda_i)$ is the confidence of $\lambda_i \in \mathcal{L}$, being a proper label of $\mathbf{x}$. The former confidence could also be formulated as an estimation of $p(\mathbf{y}|\mathbf{x}) : \mathbf{y} \in \mathcal{Y}$ supported only on $\mathbf{y}$ satisfying $\sum_{n=1}^{|\mathcal{Y}|} \mathbf{y}_{(n)} = 1$. Meanwhile the latter is an estimation of $p(y^i|\mathbf{x}) : y^i \in [0, 1]$ (*i.e.* $p(y^i = 1|\mathbf{x})$ or $p(\lambda_i|\mathbf{x})$). To keep the notation uncluttered, we use $s^{\mathbf{y}}(\mathbf{x}))$ to denote $s(\mathbf{x}, \mathbf{y})$; and $h^i(\mathbf{x})$ (respectively $s^i(\mathbf{x})$) to denote $h(\mathbf{x}, \lambda_i)$ (respectively $s(\mathbf{x}, \lambda_i)$). These different multi-label outputs are formulated as :

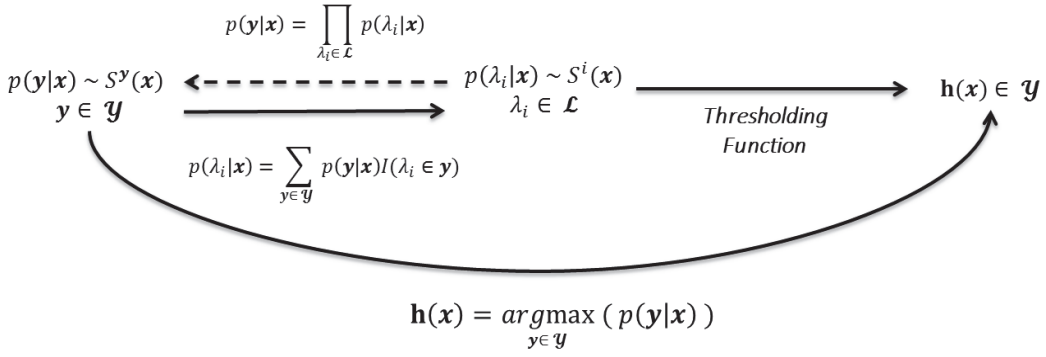
- A bi-partition of the label space $\mathcal{L}$ into relevant and irrelevant labels: $\mathbf{h}(\mathbf{x}) = (h^1(\mathbf{x}), \cdots, h^q(\mathbf{x})) \in \mathcal{Y} \subseteq \{0, 1\}^q$.
- A label probability score vector, where the vector component indicates the relevance of the label $\lambda_i \in \mathcal{L}$: $\mathbf{s}(\mathbf{x}) = (s^1(\mathbf{x}), \cdots, s^q(\mathbf{x})) \in [0; 1]^q$.
- A probability score for each labelset indicating the relevance of each possible label combination : $\forall \mathbf{y} \in \mathcal{Y}\ s^{\mathbf{y}}(\mathbf{x}) \in [0; 1]$.

In practice, the choice of the output space depends on the application context. For example, in fully automated annotation the prediction model must be as accurate as possible in its labels assignment. Such context matches to the email transfer system which forwards an incoming email to all relevant departments of a company. Besides, the predicted label's score reflects the confidence degree of the model to associate $\mathbf{x}$ with the label λ. Thus, its allows ranking a set of labels regarding their appropriateness for the predicted instance, and reciprocally, allows ranking a set of instances regarding their appropriateness for the label λ. Of course, it may be that in some situations the label probability score and the label bi-partition space are both important for decision making.

2.2.1 Multi-Label output transformations

In several multi-label tasks, the output space of the most suitable algorithm is not adequate to the application needs or not adequate to the objective loss function. Hence, the need to transform the predictions, via a mapping function $\mathcal{M}$, to meet the adequate output space. The most popular transformation is to switch the label probabilities to label space bi-partition. Meanwhile, it is also possible to transit from labelset probabilities to a vector of label probabilities and vice-versa (under particular hypothesis). Figure 2.1 summarizes the possible outputs transitions that we will examine in the flowing subsections.

FIGURE 2.1: Multi-Label output transformations



Transition from $s^{\mathbf{y}}(\mathbf{x})$ to $s^{i}(\mathbf{x})$

In this case, we consider that the multi-label model provides an estimation of the probability distribution over all possible labelsets $s^{\mathbf{y}}(\mathbf{x}) \ : \ \mathbf{y} \in \mathcal{Y}$; meanwhile, the desired output space is the label probability score (i.e. a vector of label probabilities scores: $\mathbf{s}(\mathbf{x}) = (s^1(\mathbf{x}), \cdots, s^q(\mathbf{x})) \in [0,1]^q$). This Transformation is carried out via a marginalization procedure over the labelsets

probabilities [19]. The transition is guaranteed since that, the labelset probability scores are an estimation of the conditional joint label distribution ($s^{\mathbf{y}}(\mathbf{x}) \simeq p(\mathbf{y}|\mathbf{x})$), and the label probability scores are an estimation of the conditional marginal label distribution ($s^{i}(\mathbf{x}) \simeq p(\lambda_i|\mathbf{x})$).

$$\mathcal{M} : [0,1]^{|\mathcal{Y}|} \rightarrow [0,1]^{q}$$

$$p(\lambda_i|\mathbf{x}) = \sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x}).I(\lambda_i \in \mathbf{y}) \simeq s^{i}(\mathbf{x}) = \sum_{y\in\mathcal{Y}} s^{\mathbf{y}}(\mathbf{x}).\mathbf{y}^{i}$$

The transformation $\mathcal{M}$ consists of simply estimating each label score $s^{i}(\mathbf{x})$ as the sum of the probabilities predicted for all labelsets containing the label λ_i. The example below illustrates the transition from $s^{\mathbf{y}}(\mathbf{x})$ to $s^{i}(\mathbf{x})$ of a possible labelset probability distribution predicted by a multi-label model.

$\mathbf{y} \in \mathcal{L}$	$s^{\mathbf{y}}(\mathbf{x})$	λ_1	λ_2	λ_3	λ_4
$\{\lambda_1, \lambda_4\}$	0.7	1	0	0	1
$\{\lambda_3, \lambda_4\}$	0.2	0	0	1	1
$\{\lambda_1\}$	0.1	1	0	0	0
$\{\lambda_2, \lambda_3, \lambda_4\}$	0.0	0	1	1	1
$\cdots$	0.0	-	-	-	-
$s^{i}(\mathbf{x}) = \sum_{\mathbf{y}} s^{\mathbf{y}}(\mathbf{x}) y^{i}$		0.8	0	0.2	0.9

$s^{\mathbf{y}}(\mathbf{x})$ is a possible labelset probability distribution provided by a multi-label model

Transition from $s^{i}(\mathbf{x})$ to $s^{\mathbf{y}}(\mathbf{x})$

In this case we consider that the multi-label model provides a probability score for each label $s^{i}(\mathbf{x}) \in [0,1]^{q} : i \in \{1; \cdots; q\}$ and the desired output space is the probability distribution over all possible labelsets $s^{\mathbf{y}}(\mathbf{x}) : \mathbf{y} \in \mathcal{Y}$. The transformation is represented by the dashed line in Figure 2.1 and is based on the assumption of conditional label independence given $\mathbf{x}$. When this condition holds [20], the joint probability estimation $p(\mathbf{y}|\mathbf{x})$ can be written as the product of the marginal probabilities $p(\lambda_i|\mathbf{x})$. Thus, the mapping function is simply formulated as :

$$\mathcal{M} : [0,1]^{q} \rightarrow [0,1]^{|\mathcal{Y}|}$$

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{q} p(\lambda_i|\mathbf{x}) \simeq s^{\mathbf{y}}(\mathbf{x}) = \prod_{i=1}^{q} s^i(\mathbf{x})$$

Such transition is based on a strong assumption that is hard to check, but useful when the multi-label classification is an intermediate task where the independence condition holds [21].

Transition from $s^{\mathbf{y}}(\mathbf{x})$ to $\mathbf{h}(\mathbf{x})$

In this case, we consider that the multi-label model provides an estimation of the probability distribution over all possible labelsets $s^{\mathbf{y}}(\mathbf{x}) : \mathbf{y} \in \mathcal{Y}$; and, the desired output is a vector of crisp labels (labelset). This Transformation is carried out via a simple selection of the labelset with the larger probability score.

$$\mathcal{M} : [0,1]^q \rightarrow \mathcal{Y}$$

$$\mathbf{h}(\mathbf{x}) = \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}}\, p(\mathbf{y}|\mathbf{x}) \simeq \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}}\, s^{\mathbf{y}}(\mathbf{x})$$

Transition from $s^i(\mathbf{x})$ to $\mathbf{h}(\mathbf{x})$

In this case, we consider that the multi-label model outputs, for each label, a probability score $s^i(\mathbf{x})$ and the desired output is a vector of crisp labels (labelset). Such transformation function is well known as thresholding procedure and commonly noted as $\tau(\cdot)$ [22–24]. The straightforward option is to implement the thresholding function $\tau(\cdot)$ as 0.5 constant for all the label score predictions. This thresholding procedure is also used to guide the multi-label model to be optimal for a particular loss metric [25]. In section 2.5.4 a discussion is given about loss guided threshold calibration in general multi-label models and in Chapter 4 and Chapter 5 we discuss how the thresholding strategy can be used in ensemble multi-label models. Thus, the mapping function is simply formulated as :

$$\mathcal{M} : [0,1]^q \rightarrow [0,1]^q$$

2.3 Multi-label evaluation metrics

The generalization performance of a multi-label model is evaluated differently from traditional single-label models. Multi-label evaluation metrics are more complicated as each instance can be associated with multiple labels simultaneously. A multi-label prediction could be partially correct (where some labels are correctly predicted), fully wrong (where all predictions are wrong),

or fully correct (where all labels are correctly predicted). For this propose, several performance metrics have been proposed in the multi-label literature [9, 26]. These metrics can be distinguished by the multi-label outputs they consider: Some metrics are specific to evaluate multi-label score outputs (probability-based metrics) while others are specific to evaluate crisp labels output (bi-partition-based metrics).

2.3.1 Bi-partition-based metrics

These metrics can also be categorized according to how they evaluate the output vectors. Tsoumakas *et al.* [3] categorize the multi-label metrics into two groups, namely : *label-wise metrics* and *instance-wise metrics*. Metrics in the first group conduct a separate evaluation for each label, then average the measures across the labels. On the other hand, instance-wise metrics are computed for each evaluated instance to be averaged, in a second time, over the test evaluation sample.

A more general and theoretical formulation of these metrics categorization was given by Dembczyński *et al.* [18]. Authors define the *label-wise decomposable multi-label loss* metrics as a category of functions where the risk minimizer is obtained by minimizing the risk over each label separately ; and the *instance-wise decomposable multi-label loss* metrics as a category of multi-label loss functions where the risk-minimizing the prediction is only obtained by minimizing the risk jointly over all labels for each instance.

Let $\mathcal{D} = \{(\mathbf{x}_{(j)}, \mathbf{y}_{(j)}), 1 \leq j \leq n\}$ be a multi-label test data set with n instances represented in the form of an input feature matrix $\mathbf{X} = [\mathbf{x}_{(1)}, \cdots, \mathbf{x}_{(n)}]^\intercal$ and an output label matrix $\mathbf{Y} = [\mathbf{y}_{(1)}, \cdots, \mathbf{y}_{(n)}]^\intercal$. Respectively, let $\mathbf{h}(\mathbf{X})$ represent the matrix of predictions. Let tp^i, fp^i, tn^i and fn^i represent the number of *true positives*, *false positives*, *true negatives* and *false negatives* for a label λ_i. Multi-label learning loss metrics evaluating the relevance of the predicted bi-partition can be formulated as :

$$L : \{0;1\}^q \times \{0;1\}^q \to \mathbb{R}^q_+$$

In the following, we give a mathematical description of the commonly used multi-label loss metrics evaluating the crisp label predictions.

- The *Subset 0/1 loss* generalizes the well-known *0/1 loss* from the traditional single-label classification to the multi-label context and defined as:

$$Subset\ 0/1\ loss(\mathbf{Y}, \mathbf{h}(\mathbf{X})) = \frac{1}{n}\sum_{j=1}^{n} \mathcal{I}(\mathbf{h}(\mathbf{x}_{(j)}) \neq \mathbf{y}_{(j)}) \tag{2.2}$$

  Where $\mathcal{I}(\phi)$ equals to 1 if ϕ holds and 0 otherwise for any predicate ϕ. The *Subset 0/1 loss* metric is known to be a very strict evaluation measure as it does not distinguish between

the "partially correct" and the "fully wrong" predictions and thus requires an exact match between the true set of labels and the predicted set of labels.

- The *Jaccard loss* is originally defined by set operators as one minus the ratio of intersection and union and formulated as follows:

$$Jaccard\ loss(\mathbf{Y}, \mathbf{h}(\mathbf{X})) = 1 - \frac{1}{n}\sum_{j=1}^{n}\frac{\sum_{i=1}^{q} y_{(j)}^{i} h^{i}(\mathbf{x}_{(j)})}{\sum_{i=1}^{q} y_{(j)}^{i} + \sum_{i=1}^{q} h^{i}(\mathbf{x}_{(j)}) - \sum_{i=1}^{q} y_{(j)}^{i} h^{i}(\mathbf{x}_{(j)})} \tag{2.3}$$

- The *Instance-F1 loss* is defined as an instance-wise metric. Its value is the average of the F1 score for each instance in the test data set.

$$Instance\text{-}F1\ loss(\mathbf{Y}, \mathbf{h}(\mathbf{X})) = 1 - \frac{1}{n}\sum_{j=1}^{n}\frac{2\sum_{i=1}^{q} y_{(j)}^{i} h^{i}(\mathbf{x}_{(j)})}{\sum_{i=1}^{q} y_{(j)}^{i} + \sum_{i=1}^{q} h^{i}(\mathbf{x}_{(j)})} \tag{2.4}$$

- The *Macro-F1 loss* is an average of the label F1 score computed separately for each label. Assuming that p^i and p^i are the precision and recall over the label i. The *Macro-F1 loss* is defined as :

$$Macro\text{-}F1\ loss(\mathbf{Y}, \mathbf{h}(\mathbf{X})) = 1 - \frac{1}{q}\sum_{i=1}^{q}\frac{2.p^{i}.r^{i}}{p^{i} + r^{i}} \tag{2.5}$$

- The *Micro-F1 loss*, in contrast to the *Macro-F1 loss*, first sums the contingency matrices (*i.e.* tp^i, tn^i, fp^i, fn^i) for all labels and then computes the F1 score as follow:

$$Micro\text{-}F1\ loss(\mathbf{Y}, \mathbf{h}(\mathbf{X})) = 1 - F1(\sum_{i=1}^{q} tp^{i}, \sum_{i=1}^{q} tn^{i}, \sum_{i=1}^{q} fp^{i}, \sum_{i=1}^{q} fn^{i}) \tag{2.6}$$

  It is important to notice that, by construction, the *Micro-F1 loss* falls neither in the label-wise category nor in the instance-wise category [18] of the multi-label metrics.

- The *Hamming loss* evaluates the accuracy as the average of the binary classification error. It measures the percentage of incorrectly predicted labels to the total number of labels.

$$Hamming\ loss(\mathbf{Y}, \mathbf{h}(\mathbf{X})) = \frac{1}{q}\sum_{i=1}^{q}\frac{\sum_{j=1}^{n} \mathcal{I}(y_{(j)}^{i} \neq h^{i}(\mathbf{x}_{(j)}))}{n} \tag{2.7}$$

By definition, these metrics take values in the interval [0; 1] and the smaller the value, the better the algorithm performance is (the best value is scored 0 and the worst at 1). As the *Subset 0/1 loss*, *Jaccard loss* and the *Instance-F1 loss* consider each instance separately they are instance-wise decomposable metrics. It is also important to highlight that these metrics are not decomposable over single labels since their risk minimizer could not be obtained by minimizing the risk

separately for each label [18]. Besides, the *Macro-F1 loss* is computed as an average of the performance over the labels. Thus it is considered as a label-wise decomposable metric. On the other hand, the *Hamming loss*, especially, could be considered, simultaneously, as a label-wise decomposable and as an instance-wise decomposable metric, since that the metric is decomposable over single instances and also decomposable over single labels [27]. However, it is generally considered, in the literature, only as a label-wise decomposable metric [18]. In this thesis, we will only focus on the label decomposition of the *Hamming loss*.

2.3.2 Probability-based metrics

When the multi-label model outputs the real-valued or probabilities scores, others multi-label metrics can be defined as well. Generally, these metrics evaluate the model performance from a ranking perspective [28] and formulated as:

$$L : \{0;1\}^q \times \mathbb{R}^q \to \mathbb{R}^q_+$$

- The *Ranking loss* is defined as the average fraction of label pairs that are reversely ordered for the prediction. It is defended as :

$$Ranking\ loss(\mathbf{Y}, \mathbf{s}(\mathbf{X})) = \frac{1}{n}\sum_{j=1}^{n} \frac{|\{(\lambda_u, \lambda_v)|s^u(x_{(j)}) \leq s^u(x_{(j)}), (\lambda_u, \lambda_v) \in l \times \bar{l}\}|}{|l_{(j)}||\bar{l}_{(j)}|} \tag{2.8}$$

  where $l_{(j)}$ denotes the set of labels associated with the instance $\mathbf{x}_{(j)}$ and $\bar{l}_{(j)}$ denotes its complementary set in $\mathcal{L}$ (*i.e.* $y^i_{(j)} = 1 \Leftrightarrow \lambda_i \in l_{(j)}$ and $y^i_{(j)} = 0 \Leftrightarrow \lambda_i \in \bar{l}_{(j)}$).

- The *One-error* evaluates the fraction of prediction where the top-ranked label is not on the set of the true relevant labels. The *One-error* is formulated as follows :

$$One\text{-}error(\mathbf{Y}, \mathbf{s}(\mathbf{X})) = \frac{1}{n}\sum_{j=1}^{n} \mathcal{I}([\underset{\lambda_i \in \mathcal{L}}{\operatorname{argmax}}\, s^i(\mathbf{x}_{(j)})] \notin l_{(j)})$$

  The *One-error* takes values between 0 and 1. The smaller the value is, the better the performance is. Note that, for single-label classification problems, the *One-error* is identical to ordinary classification error.

- The *Coverage* evaluates the number of required steps, on average, to move down the ranked predicted scores to cover all the labels associated with the instance. The *Coverage* is formulated as follows :

$$Coverage(\mathbf{Y}, \mathbf{s}(\mathbf{X})) = \frac{1}{p}\sum_{i=1}^{p} \max_{\lambda \in l_{(j)}} [rank(\mathbf{s}(\mathbf{x}_{(j)}))] - 1$$

For either *One-error*, *Coverage* and *Ranking loss*, the smaller the metric value the better the model's performance, where 0 is the optimal value for the *Ranking loss* and the *One-error* whereas the optimal value of the *Coverage* is $\frac{1}{n}\sum_{j=1}^{n} |\mathbf{y}_j| - 1$.

Furthermore, most of the classical single-label performance metrics can also be generalized to the multi-label context and used to evaluates the quality of the predicted scores via a *Macro* or *Micro* averaging process, as for the *Macro-F1 loss* and the *Micro-F1 loss*. Thus similarly to these two metrics Zhang and Zhou defined the multi-label AUC metric such as the AUC Area Under ROC Curve [28], where the AUC_{macro} is the averaged value of the AUC across all the labels.

2.4 Multi-label learning methods

Basically, existing multi-label learning methods may be grouped into two main approaches: *algorithm adaptation* and (b) *problem transformation* [3, 9]. *Algorithm adaptation* approaches extends specific learning algorithms to handle the multi-label data directly. *Problem transformation* approaches, on the other hand, comprises approaches that transform the multi-label learning problem into either one or more traditional single-label learning problems. The single-label learning problems are then solved with a commonly used single-label classification approach. Finally, and the output predictions are transformed back to the multi-label representation.

2.4.1 Algorithm adaptation approaches

In this category of multi-label models almost all traditional paradigms in conventional single-label classification have been revisited to be adapted to handle multi-labeled data. Models in this category are based on existing algorithms such as: classical decision trees algorithm [6], Support Vector Machines (SVMs) [29], neural networks [8] and k-nearest neighbours (K-NN) [5]. The keys concerns in these models are *i)* how to deal with the label overlap and *ii)* how to consider the links (correlation) among different labels while improving the prediction quality.

Algorithms based on Decision Tree

Due to their hierarchical outcome and their interpretability, decision trees have been widely used in multi-label models especially in genomic applications [6, 7, 30–32].

Clare *et al.* modified in [6] the entropy function in the classical decision tree C4.5 algorithm to handle instances associated with multiple labels. In this new multi-label C4.5 algorithm (termed ML-C4.5), multiple labels in the tree's leaves are allowed, and the entropy formulation is adapted to quantify the information needed to describe the labels associated with instances. Formally, the modified function of entropy, for a given data set $\mathcal{D}$, sums the entropies for each individual label q and considers both the membership and the non-membership of labels as :

$$entropy(\mathcal{D}) = -\sum_{i=1}^{q}(p(\lambda_i)\log p(\lambda_i) + q(\lambda_i)\log q(\lambda_i))$$

Where $p(\lambda_i)$ is the probability (relative frequency) of the label λ_i and $q(\lambda_i) = 1 - p(\lambda_i)$.

By adopting the rule of maximum information gain, which is the difference of entropy after splitting, the decision tree is equipped to handle the multi-label data directly. Finally, the leaves of the tree are allowed to predict the most frequent set of labels in the branch.

In [32], Kocev *et al.* propose the *Predictive Clustering Tree* (PCT), which considers the decision tree as a hierarchy of clusters where multi-labeled data is partitioned [7]. The induction process in PCT is a top-down generation of clusters where the intra-cluster variation is minimized. The model can be assimilated to a hierarchy of clusters where nodes are partitioned into smaller clusters by traversing from top to bottom, and each leaf is labeled with its cluster's prototype in the prediction step. The idea behind the PCT model is to provide the possibility to adopt of a variance function describing the nodes and a prototype function to decide over their values. In the multi-label setting, the PCT uses the sum of the Gini indices as a variation criterion in order to consider the links between the labels. The variance function is formulated as:

$$Var(\mathcal{D}) = \sum_{i=1}^{q} Gini(\mathcal{D}, \mathbf{Y}^i), Gini(\mathcal{D}, \mathbf{Y}^i) = 1 - (p^2(\lambda_i) + q^2(\lambda_i))$$

Where $p(\lambda_i)$ is the probability of the label λ_i and $q(\lambda_i) = 1 - p(\lambda_i)$.

Algorithms based on SVM

Support Vector Machines SVMs have been wildly used in the multi-label context. they generally construct a tailored model to minimize an objective loss function explicitly [33–35].

For instance, in [29] Elisseeff and Weston presented a ranking approach based SVM to handle multi-label data termed RankSVM. The proposed method tries to control the model complexity while minimizing the empirical error. But, the key idea of the RankSVM algorithm is to use the *Ranking loss* as a specific loss function in the inner optimization process and thus allows the

model to capture multi-label characteristics of the multi-label task. In [36, 37] authors propose to extends the structural SVMs to minimize the *Hamming loss*.

The SVMs have also been used in order enhance the classification performance of exiting multi-label models by constructing a new kernel that expresses the correlation among different labels [38]. Furthermore, in [39] authors introduced a generalization of SSVMs that can be implemented for optimizing a variety of multi-label loss metrics.

Algorithms based on Probabilistic Framework

Many of the approaches to multi-label learning mainly rely on discriminative modeling techniques; nevertheless, some generative models have also been devised. In [40, 41] a probabilistic generative model for multi-label document classification were presented. The proposed approach is constructed to model multiple labels associated with each input document. The model assumes that a document is generated by a mixture of word distributions, where each word distribution is a label. In the learning step the *expectation maximisation* is used to estimates the mixture weights and the word distribution. While, in the prediction step, the Bayes rule is applied to predict the most probable set of labels given the document as an input. Confronted to other multi-label learning models, these probabilistic models can only be applied for text classification. More general approaches are desirable to handle a wider range of multi-label learning tasks.

Algorithms based on Neural Networks

Zhang and Zhou proposed to use the neural networks in [8], and present *Back-Propagation Multi-Mabel Mearning* BP-MLL; which is an adaptation of the traditional multilayer feed-forward neural network in the multi-label learning. The important modification of the algorithm is the use of a function error that considers multi-labeled data and closely related to the *Ranking loss*. The idea is to assume that the labels associated with an instance should have a higher ranked than those not associated with the instance. The neural network is trained with gradient descendent algorithm where the minimized error is formulated as :

$$\mathcal{E} = \sum_{i=1}^{n} \frac{1}{|\mathbf{y}_{(i)}||\bar{\mathbf{y}}_{(i)}|} \sum_{(j,k)\in \mathbf{y}_{(j)}\times\bar{\mathbf{y}}_{(k)}} exp(-(f^{j}_{(i)}(\mathbf{x}_{(i)}) - f^{k}_{(i)}(\mathbf{x}_{(i)})))$$

where $(f^{j}_{(i)}(\mathbf{x}_{(i)}) - f^{k}_{(i)}(\mathbf{x}_{(i)})$ measures the difference between the outputs of the network on the set of relevant labels and the set of irrelevant ones for the i-th instance.

Besides, inspired by the *Radial Basis Function* (RBF) methods [42] Zhang[43] proposed the Multi-Label Radial Basis Function (ML-RBF). The proposed network is trained in a two-stage

procedure. In the first stage, the basis functions of the hidden layer are learned through a k-means instances clustering aver each label. The stage aims to construct the prototype vectors of the first-layer basis functions as the centroids of the clusters. In the second stage, the second layer's weights are optimized via the minimization of the sum-of-squares error function. The link between the labels is considered via a connection between all the basis functions corresponding to the prototype vectors of all labels.

Algorithms based on k-Nearest Neighbor

In [5] Zhang and Zhou extend the k-Nearest Neighbor (kNN) to handle multi-label data. The central idea of ML-kNN is to label each instance based on the labels of the neighboring instances. Although the determination of the labels for a new test instance is different, the algorithm uses the prior and the posterior probabilities of each label among the kNN. The statistical information is gained from the labelsets of the neighboring instances via the *Maximum A Posterior* to predict the labels of a new example. Formally, given an unseen example $\mathbf{x}$, the algorithm first determines the set of k nearest neighbors: $N = \{(\mathbf{x}_{(i)}, \mathbf{y}_{(i)}) | 1 \leq i \leq k\}$, and gets a vector counting the number of instances associated to each label in the neighborhood of $\mathbf{x}$: $c = (c^1, \cdots c^q)$ where $c^j = \sum_{(\mathbf{x}_{(i)}, \mathbf{y}_{(i)}) \in N} \mathcal{I}(\mathbf{y}^j_{(i)} = 1)$. Then, based on the prior and the posterior probabilities of each label within the neighborhood, the algorithm identifies the labelset to be associated with the new instance via the maximum a posteriori principle:

$$\mathbf{y}^i = \begin{cases} 1 & if\ p(c^j|\mathbf{y}^j = 1)p(\mathbf{y}^j = 1) \geq p(c^j|\mathbf{y}^j = 0)p(\mathbf{y}^j = 0) \\ 0 & otherwise \end{cases}$$

2.4.2 Problem transformation approaches

The straightforward strategy to handling the multi-label learning task is converting it into one or a series of mono-label learning tasks where conventional mono-label learning models can be applied directly. The key principle is to get rid of the label overlap in the original target space. Compared to the *algorithm adaptation approach*, the *problem transformation approach* is more flexible since any conventional mono-label model can be used. First, the original multi-label task is transformed into one or more single-label tasks solved via traditional algorithms. Then in the prediction step, the outputs transformed back into the initial representation. *Problem Transformation approaches* can be grouped into three schemes: *i) Binary Relevance* (BR), *ii) Pair-wise*, and *iii) Label Power-set* (LP).

Binary Relevance methods (BR)

The main idea in the *Binary Relevance* scheme is to switch the multi-label problem into several distinct binary problems. BR learns $q = |\mathcal{Y}|$ binary models; each specialized in one label independently from the others. Concretely, for each label λ_i, the associated model learn to predict, as positive instances, all the training samples associated with λ_i and the remaining samples are considered as negative: $h^i : \mathcal{X} \longrightarrow \{0, 1\}$, where h^i, is the binary model associated to λ_i. For an unseen instance $\mathbf{x}$, the predicted labels $\hat{\mathbf{y}}$ are the concatenation of the predictions trough all the binary models *i.e.* $\hat{\mathbf{y}} = (h^1(\mathbf{x}), \cdots, h^q(\mathbf{x}))$.

The BR style models are intuitive approaches, easy to implement, with a low computation complexity. Furthermore, it can also be combined with many binary learning algorithms such as *Support Vector Machines* and Artificial Neural Networks or KNN [4] and has been widely used as a baseline to evaluate the performance of multi-label learning models [4]. Nevertheless, the main drawback of the binary relevance scheme is its hard label dependence assumption. Indeed, as each label is treated independently from the others, the BR approach does not consider any links among the labels. It also suffers from the target imbalance problem due to the typical sparsity of labels in multi-label data sets. Indeed, for each label, the number of positive instances can be significantly less than the number of negative instances. Furthermore, when dealing with a large number of labels, the BR scheme may not scale since a binary model has to be constructed for each label.

In order to take into account the label links, several works propose to overcome the BR label independence assumption.

Following the one-versus-one philosophy Hüllermeier *et. al.* [44] propose to transform the original multi-label problem into $(q-1) \times q/2$ binary tasks, one for each pair of labels. In each task, example associated to one of the labels are considered as positive while instances belonging to both labels or any label are not considered as training samples. So a binary model is used to discriminates the two labels. In the prediction step, the vote of the $(q-1) \times q/2$ classifiers gives a ranking of labels according to the predictions of the binary models.

However, even if the pairwise model takes into consideration pairwise links between the labels, it predicts only label ranking and is not able to output a bi-partition of the label space. To do so, an other variation has been presented in [45, 46] termed *Calibrated Label Ranking* (CLR) incorporating a strategy to ameliorate the selection of relevant labels. The idea is to introduce, an artificial label λ_0, which act identically to a BR transformation and serves as a split point separating the relevant labels from the irrelevant ones. Even though these methods consider the links between pairs of labels, which is relatively effective. Links between labels are generally grouped on more than two labels. Furthermore, these models have a significant complexity in the prediction step and require consulting all the generated binary models, which may be impractical

for large labeled data. To speed up prediction step several voting schemas has been proposed aiming to avoid evaluating all pairwise classifiers [47, 48]. Still remain, however, the need to store a quadratic number of binary models.

Classifier Chains algorithm (CC)

To tackle the BR label independence assumption Read *et al.* [17] proposed the multi-label *classifier chain* (CC). The main idea of CC is to generates q binary models linked in such a way that the input space of each binary model is extended with the 0/1 labels associations of all its previous classifiers. Specifically, each model learns a mapping from $\mathcal{X} \times \{0,1\}^{i-1}$ to $\{0,1\}$ reflectively for each label λ_i as :

$$\begin{aligned} h^i : \mathcal{X} \times \{0,1\}^{i-1} &\rightarrow \{0;1\} \\ (\mathbf{x}, y^1, \cdots y^{i-1}) &\rightarrow p(\mathbf{y}^i|\mathbf{x}, y^1, \cdots y^{i-1}) \end{aligned} \tag{2.9}$$

The label models h^i can also be interpreted as a probabilistic classifier where the predictions are an estimation of the probability of $y^i = 1$. From this perspective, the CC model can exploits the probability product rule to estimate the joint probability distribution $p(\mathbf{y}|\mathbf{x})$. According to the chain rule, the joint probability can be decomposed into a product of conditionals probabilities: $p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}^1|\mathbf{x}) \times p(\mathbf{y}^2|\mathbf{x}, \mathbf{y}^1) \times \cdots \times p(\mathbf{y}^q|\mathbf{x}, \mathbf{y}^1, \cdots, \mathbf{y}^{q-1})$. Thus the CC model considers the links between the labels effectively and overcomes the label independence assumption of BR.

In their first proposition, Read *et al.* [17], suggest to classify the labels in a greedy sequence where the each is decided by maximizing $p(\mathbf{y}^i|\mathbf{x}, \mathbf{y}^1, .., \mathbf{y}^{i-1})$ directly in each step. Despite, this procedure has three shortcomings: First, the predicted subset of labels can be different from the real mode of the distribution. Second, in the prediction step, the error prediction can be spread to the following labels predictions. Third, the global label prediction depends on the order used to chain the binary models.

To deal with the first and the second issues, a Bayes optimal approach of forming classifier chains based in probability theory, termed Probabilistic Classifier Chains (PCC), was proposed in [49]. PCC tests all possible chain order and predicts the new set of label as $\hat{y} = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}}, p(\mathbf{y}|\mathbf{x})$. Despite obtaining better performance than CC, as PCC has to look at each of 2^q possible labelset in the prediction stage. Thus, the exact inference can become impractical and the model applicability is only advisable for tasks with a fair number of labels ($q \leq 15$).

To avoid the exhaustive search -while bypassing a greedy one-, approximation techniques may have to be used to cope with the computation complexity. Several variant has been proposed with some more accurate search, such as approximate search [50], A* search [51], or Beam Search [52].

Finally, to avoid the adverse effects of the chaining order, Real et al. [17] proposed to combine several chaining in an Ensemble of Classifier Chains (ECC). The main idea is to select the most probable label according to the prediction of several CC each based on a different order (See Section 3.3.2.2). An other strategy is proposed in [53] aiming also to increment the feature space of the BR model with labels. In this case -the (BR+) approach- for each binary classifier the feature space is augmented with $q-1$ descriptive features corresponding to the all other labels. In the prediction stage, the $q-1$ augmented features of the unlabeled instance are replaced by the prediction of BR classifier trained with the original training data.

Label Powerset algorithm -approach- (LP)

The Label Powerset (LP) approach considers each label subset as distinct meta-class. Thus it reduces the multi-label label task into one multi-class mono-label task. The transformed target represents all possible distinct subsets of labels present in the initial multi-label problem. So in the learning step, any conventional multi-class learning model can be used $h : \mathcal{X} \longrightarrow \mathcal{Y}$. When a new instance is presented, The LP outputs a class, which is actually a labelset in the original multi-label task. By combining all the labels into a single meta-class, LP is also able to consider the links between the labels and model their correlations in the training data. Although, after the transformation step it is possible to have a restricted number of training instances for the less frequent labelsets, creating a class imbalance issue. Besides, the LP approach only considers the distinct labelsets in the training data, so it is not able to predict unseen labelsets [17]. Another limitation of the LP scheme is the potentially large number of classes to be handled in the multi-label format, in the worst-case exponential with the number of labels $|\mathcal{L}| = 2^q$ [3].

In order to bypass these shortcomings, a Pruned Problem Transformation named *Pruned Sets* (PS), has been developed by Read et al. [16]. PS extends the LP transformation scheme while avoiding both its complexity problems and unbalanced class representation. The main idea of PS is to prune examples with less frequent labelsets to withdraw the LP complexity. To make up for the loss of information in the pruning step, the model reintroduces the pruned sample associated with the frequent subset of their original labelset. Finally, to output labelsets outside the training set, Read et al. [16] propose to build a committee of PS models where the prediction is based on label vote (see Section 3.3).

In a similar perspective to reduce the complexity of the LP model, Tsoumakas et al. [54] addressed the LP complexity through the HOMER algorithm for *Hierarchy Of Multi-label classifiERs*. The main idea, behind of the HOMER algorithm, is to convert the original multi-label task into a tree hierarchy of reduced multi-label problems, each dealing with a small number of labels. At each node in the hierarchy, the label space $\mathcal{L}$ is clustered -using a clustering algorithm such as k-means- into balanced groups of similar labels, considered as meta-label. Then,

a multi-label model is adopted to predict one or more meta-labels. The HOMER algorithm is a computationally efficient multi-label model, especially for large multi-labeled data sets with a linear complexity in the training step and a logarithmic complexity in the testing step (with respect to the size of the label space $|\mathcal{L}| = q$).

2.4.3 Algorithm adaptation and problem transformation in practice

In practice, the use of *algorithm adaptation* or *problem transformation* approach depends on the application needs and the user's preferences. Algorithm adaptation approaches have the convenience to extend the scope of the well-known learning technique, and hence their use is more generic. Moreover, multi-label models in this category have the advantage to be well designed for specific application domains such as the text classification [8, 40, 45]. On the other hand, problem transformation approaches are superior regarding their simplicity and generality. They have the advantage to be model-free and are considered as meta-learners. By adopting models in this category, all well-known and efficient algorithms in machine learning can be used and applied to any domains of multi-label classification. Additionally, the effects produced by the transformation step can be relieved by simple schemes. For instance, the problem of imbalanced training data can be alleviated by the under or over sampling strategies.

The efficiency of a multi-label model is also challenged by the dimensionality of the label space. On one side, the cost of training a multi-label model, in term of computation, may be affected by the number of labels. Simple algorithms such as Binary Relevance have linear complexity on q, but algorithms that involve a pairwise confrontation between the labels have a worse training cost, *e.g.*, pair-wise methods [44, 45]. Even more, the complexity of the LP model is exponential with the number of the label since that the learned multi-class model has an exponential complexity. Besides, the prediction step is also influenced by the number of models which can be time-consuming [47]. Also, the memory requirements represent an additional important factor [2, 44]. In fact, these important factors need to be considered simultaneously when developing a new multi-label model in order to gain time and space efficiency. Besides, algorithm adaptation approaches also incur considerable algorithmic complexity [55]. On the other hand, they offer the possibility to be tailored to the application context, for example, by adding constraints over the feature space and the label representation especially for domains such as in text classification where the labels can be organized hierarchically. Alongside with the two multi-label models categories, Madjarov *et al.* [9] distinguish a third category of multi-label models based on top of algorithm adaptation and problem transformation models. This third category will be discussed in detail in Chapter 3.

For completeness, it is also important to note that multi-label models can also be categorized based on the considered order of correlations [28]. Three categories are distinguished *First-order*

models, *Second-order models* and *High-order models*. The First-order models ignore the label dependencies and decompose the multi-label tasks into a set of independent binary problems as in the BR model. Second-order models consider the pairwise correlation between the labels such as CLR [45, 46] or QWeighted approach to multi-label learning [47]. Finally, High-order models consider the high-order links between the labels with stronger correlation-modeling capabilities such as LP and CC [17].

Besides multi-label classification, another popular problem in multi-label learning is *label ranking* which learns an ordering of the labels based on their relevance to a given instance. In this thesis, we mainly focus on multi-label classification. More details on connections between multi-label classification and label ranking can be found in [2].

2.5 Multi-label classifiers and loss minimization

To meet the needs of the multi-labeled tasks, the multi-label algorithms should be able to consider a multitude of loss metrics. To do so, algorithms adopt two possible approaches: The first one is to model the entire distribution of $\mathbf{y}$ given $\mathbf{x}$ then use the loss formulation (2.1) to give the optimal prediction for any loss. The second approach is to model directly a function giving the optimal prediction for the objective loss. In the latter case, the choice of the loss function is made before the model construction and the purpose of the learning algorithm is to learn from the training examples by explicitly or implicitly optimizing the specific metric [33, 34]. However, since those multi-label metrics are generally neither convex nor differentiable, constructing an optimal predictor that optimizes directly the cost function is not straightforward. Hence, the standard approach consists in minimizing a convex surrogate rather than the original loss metric [56].

2.5.1 Label Dependence in multi-label Learning

The idea of taking advantage of the label dependence to enhance the performance of multi-label predictions intuitively makes sense when it is compared to the BR base-line predictions which ignore the mutual labels links. But, the comprehension of this nature of the possible link between labels is only recently taking shape. Authors of [18, 20] give a theoretical perspective of the multi-label classification pointing out from a probabilistic basis, the difference between
- marginal dependence: where $p(y^j|y^k) \neq p(y^j)$; and
- conditional dependence; where $p(y^j|y^k) \neq p(y^j|y^k, \mathbf{x})$.

The conditional dependence between the labels and the feature can be expressed in terms of graphical models such as the conditional random fields in [57]. Indeed, these models provide the possibility to represent the relationships between labels and features of a given task.

When the nature of the dependence is known in advance, using the graphical structure for modeling and learning seems to be the most appropriate solution. The output of this category of models is an estimation of the entire joint distribution of the labels. The learning cost depends primarily on the complexity of the modeled structure. Much more restrictive, however, is the inference using the joint distribution as the exact inference can become impractical. However, in these methods, the inference from the estimated joint distribution limits the applicability of this category of models to a moderate number of labels ($q \leq 15$).

Besides, by learning labels and inputs together, methods such as the LP and the CC can model the conditional dependence. In contrast, BR approach does not take any kind of label dependence into account, neither conditional or marginal. An enhancement over BR model can also be achieved using a prior understanding about the marginal dependence within the labels. Approaches like Staking (BR+) [53] tries to take advantage of the similarities between the labels and exploit the label dependence. The general scheme of these methods can be expressed as follows:

$$\mathbf{y} = \phi(\mathbf{h}(\mathbf{x}), \mathbf{x})$$

The idea is to replace the original predictions, learned separately, by adjusting using the information regarding the predictions of the other labels using a new multi-label model ϕ as a meta-model. This transformation of the initial predictions is presented as a regularization procedure or as a feature expansion strategy. This approach is used in practice to enhance the predictive performance of the BR classifier [53].

The final meta-classifier ϕ can be trained either on the BR predictions $h(\mathbf{x})$ alone or use both the predictions $h(\mathbf{x})$ and the original features x as additional inputs. It is also possible to use the score provided by the inner learning model in the BR rather than its crisp label predictions.

2.5.2 Optimality in multi-label learning

Basically, the purpose of a learning model is to minimize the expected risk of $\mathbf{h}$ with regard to the underlying joint distribution $\mathbb{P}(\mathbf{X}, \mathbf{Y})$, i.e.,

$$R_L(\mathbf{h}) = \mathbb{E}_{\mathbf{X},\mathbf{Y}}[L(\mathbf{Y}, \mathbf{h}(\mathbf{X}))]$$

A risk-minimizing model $\mathbf{h}^*$ for the loss L is determined by :

$$\mathbf{h}^* = \underset{\mathbf{h}:\mathbf{x}\to\mathbf{y}}{\operatorname{argmin}}\, \mathbb{E}_{\mathbf{Y},\mathbf{X}\sim\mathbb{P}}[L(\mathbf{h}(\mathbf{X}), \mathbf{Y})] = \underset{\mathbf{h}:\mathbf{x}\to\mathbf{y}}{\operatorname{argmin}}\, \mathbb{E}_{\mathbf{X}\sim\mathbb{P}}[\mathbb{E}_{\mathbf{Y}\sim\mathbb{P}(\mathbf{Y}|\mathbf{X})}[L(\mathbf{h}(\mathbf{X}), \mathbf{y})]]$$

Given that $\mathbf{h}^*$ optimizes the expected loss regarding the conditional distribution $p(\mathbf{Y}|\mathbf{x})$ at each given value of $\mathbf{x}$; investigating the optimal predictions at a given $\mathbf{x}$ is sufficient. Therefore, the pointwise risk-minimizing model $\mathbf{h}^*(\mathbf{x})$ is given by :

$$\mathbf{h}^*(\mathbf{x}) = \underset{\mathbf{h}:\mathbf{x}\to\mathbf{y}}{\operatorname{argmin}} \mathbb{E}_{\mathbf{Y}\sim\mathbb{P}(\mathbf{Y}|\mathbf{X})}[L(\mathbf{h}(\mathbf{X}), \mathbf{y})] \tag{2.10}$$

However, the optimization problem in 2.10 mostly requires a search over all possible binary vectors of length q. Thus, seeking an optimal solution or constructing a consistent model can be intractable depending on the loss metric and the joint distribution. The choice of the loss function depends on the application task and the metrics to measure the achievement of the required objectives [58]. Clearly, if the loss metric is instance-wise decomposable (decomposable over instances), such as the *Subset* $0/1$ *loss* or the *Hamming loss*, a consistent estimator of the optimal solution can be reached through empirical risk minimization. Nevertheless, the principal difficulty in the analysis of the multi-label loss metrics is that they are often non-label-wise decomposable,*i.e.*, the loss on predicting a vector of labels does not decompose into the sum of losses over the individual labels. Even more, multi-label metrics are generally complex, either convex or differentiable. Consequently, constructing an optimal predictor that optimizes the cost function directly is not straightforward.

Besides, for the label-wise decomposable metrics the risk-minimizing prediction can be obtained from the label marginal distributions alone, *i.e.*, $p(\mathbf{y}^i|\mathbf{x})$ [56]. However, these loss functions do not require having the joint label distribution to get the risk-minimizing predictions. This suggests that instead of modeling the joint label distribution to be marginalized over the labels, one can directly use a separate model for each label in order to estimate the required marginal distributions.

Thus, it is evident in case of the *Hamming loss* 2.7, where the risk minimizer is obtained directly from the marginal distribution h^* which is formulated as follows:

$$\mathbf{h}^*(\mathbf{x}) = \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmin}} \prod_{i=1}^{q} p(\mathbf{y}^i|\mathbf{x})$$

or equivalently, via a separate *argmax* decision rule over each label:

$$\mathbf{h}^*(\mathbf{x}) = (h^{*i}(\mathbf{x}), \cdots, h^{*q}(\mathbf{x})) \text{ where } \mathbf{h}^{*i}(\mathbf{x}) = \underset{y^i\in\{0;1\}}{\operatorname{argmax}} p(y^i|\mathbf{x})$$

For the *Macro-F*1 *loss* it has been also demonstrated that, under the conditional label independence [59], the optimal solution for 2.10 is simply obtained by sorting the probabilities over each label and setting to 1 the k-top instances and the remaining to 0.Thus, one only requires to estimates the marginal label distribution to compute the optimal predictions. Similarly, in the case of

the *Ranking loss*, one may simply use any single label model that estimates the labels' marginal probabilities thoroughly. From this simple fact, it is clear that considering only the marginal distribution $p(\mathbf{y}^i|\mathbf{x})$ is enough to minimize a loss metric that is label-wise decomposable [18].

In stark contrast, for metrics that are instance-wise decomposable but not label wise-decomposable (such as *Subset* $0/1$ *loss*, *Jaccard loss*, *Instance-F*1 *loss*), the construction of an optimal model, requires the estimation of the label joint distribution given the input. Especially, the risk-minimizing prediction for the *Subset* $0/1$ *loss* is given by the distribution mode:

$$\mathbf{h}^*(\mathbf{x}) = \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}} \sum_{t=1}^{T} p(\mathbf{y}|\mathbf{x})$$

Thus, to get the risk-minimizing prediction for the *Subset* $0/1$ *loss*, the optimal model will necessarily require the entire distribution of $\mathbf{y}$ given $\mathbf{x}$, or at least sufficient information to identify the mode of this distribution.

For the *Jaccard loss* and the *Instance-F*1 *loss*, it is an open question to determine if a closed-form solution for the risk minimizers exists or not. These two metrics are complex, and there is no simple approach to build a classifier minimizing them directly [18]. The minimization (and even the evaluation) of these two metrics is not straightforward and involves exponential-time computation, even when dealing with known label distribution [58]. Recently, Dembczyński *et. al.* [60] showed that the *Instance-F*1 *loss* can be minimized efficiently using q^2 parameters of the labels joint conditional distribution . For the *Jaccard loss*, the exact optimization is much harder [61].

In the light of the recently published theoretical results, the necessary computation for the optimal predictions can be considerably simplified using a rigorous implementation [58, 60, 62]. Nagarajan *et al.* [58] proposed an algorithm that runs in $O(q^3)$ time for a general multi-label loss metric. For specific metrics such as the *Instance-F*1 *loss* and the *Jaccard loss*, the optimum can be reached in $O(q^2)$. In [60] Dembczyński *et al.* point out that for the *Instance-F*1 *loss*, and an arbitrary distribution, the optimal solution to 2.10 can be obtained only in a quadratic number of parameters of the joint distribution $O(q^2)$. However, for the case of the *Subset* $0/1$ *loss* the risk minimizer is the mode of the joined distribution which is infeasible to estimate for arbitrary p.

For general multi-label losses, the standard approach is to employ structural support vector machines to optimize a convex upper bound for the expected loss on the training data [34, 35]. However, Dembczyński *et al.* [63] showed that the approach suffers from inconsistency, in the case of the *Instance-F*1 *loss* metric, for an arbitrary label distribution p.

2.5.3 Loss minimization in multi-label classifiers

As aforementioned, BR is the most simple and intuitive approach for the multi-label classification. It reduces the multi-label tasks into separate binary classification where the learning is conducted independently for each label. In doing so, the BR approach is based on the label independence assumption $p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{q} p(\mathbf{y}^i|\mathbf{x})$ which may be too strong, as labels are likely to be dependent in practice. Thus, the decision rule conducted by the BR model to predict a label vector is given by:

$$\mathbf{h}_{BR}(x) = \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmin}} \prod_{i=1}^{q} p(\mathbf{y}^i|\mathbf{x})$$

The BR model is not able to reach the risk-minimizing predictions for the non-label-wise decomposable metrics like $Subset\ 0/1\ loss$. But, it evidently yields the risk-minimizing predictions for the $Hamming\ loss$. More generally, if the base learner provides the estimation of the marginal label distribution $s^i(\mathbf{x})$ it it can yield the risk-minimizing predictions for a label-wise decomposable metrics [18]. Therefore, it is not reliable to criticize BR for its lack of considering links between the labels, especially when its performances are evaluated on label-wise decomposable metrics.

In contrast, as the prediction of the joint label distribution mode is equivalent to predicting most probable meta-class in the LP model, the approach is suitable for the $Subset\ 0/1\ loss$. However, in the literature, it is often defended to be the most appropriate approach for the multi-label classification tasks, as it takes into account the label dependence in the learning process. This argument is incorrect since that LP usually fails for label-wise decomposable loss functions like $Hamming\ loss$. Furthermore, it is obvious that a risk minimizer model cannot be optimized simultaneously for different multi-label loss functions. Recent theoretical studies show that multi-label classifier minimizing the $Subset\ 0/1\ loss$ would perform poorly if evaluated regarding the $Hamming\ loss$, and vice-versa [18, 64]. Nevertheless, in some (not necessarily extreme) conditions, the $Hamming\ loss$ and the $Subset\ 0/1\ loss$ risk-minimizing predictions coincide which leads to some misleading observation over the experimental results. These conditions has been characterized by Dembczyński *et al.* [18] through the following proposition.

Proposition 1. (Dembczyński *et. al.* [18])
The $Hamming\ loss$ (HL) and subset $Subset\ 0/1\ loss$ ($0/1$) have the same risk minimizer, *i.e.*, $\mathbf{h}^*_{HL}(\mathbf{x}) = \mathbf{h}^*_{0/1}(\mathbf{x})$, if one of the following conditions holds:

(1) Labels $\mathbf{y}^1, \cdots, \mathbf{y}^q$ are conditionally independent, *i.e*, $p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{q} p(\mathbf{y}^i|\mathbf{x})$.
(2) The probability of the mode of the joint probability is greater than or equal to, *i.e*, 0.5 $p(\mathbf{h}^*_{0/1}(\mathbf{x})|x) \geq 0.5$.

It is also worth to notice that, the LP approach is theoretically able to deliver an estimation of the joint distribution if its inner multi-class learner is a probabilistic model. Practically, however, the

large number of possible label sets turns out the probability estimation to a notably challenging problem. To this end, most of the LP's implementations typically do not take into consideration the labelsets outside the training set or set to 0 their probabilities [16]. This method is suitable as a trade-off between efficiency and accuracy since that to minimize the $Subset\ 0/1\ loss$ only the most probable labelsets is required.

The other possibility to bypass the complexity problems is to address the problem of predicting the set of labels in a step-wise mode (label by label) as formulated in the product rule decomposition and conducted by the CC model :

$$p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}^1|\mathbf{x}) \prod_{i=1}^{q} p(\mathbf{y}^i|\mathbf{y}^1, \cdots, \mathbf{y}^{i-1}\mathbf{x})$$

The approach breaks down the multi-label problem into a set of binary classification task models such as CC, and its variants seem to behave like the BR approach. Despite, the estimation produced by the chain model is closer to LP than that of BR estimation. Thus, it is not very clear what is the cost function optimized by CC. In [50], a deep analysis about CC optimality shows that regret of CC is quite important respectively for the both $Subset\ 0/1\ loss$ and $Hamming\ loss$ but with a lower worst-case regret for the $Subset\ 0/1\ loss$. Indeed, by selecting successively the most probable label based on each binary classifier CC, it is generally considered as a simple greedy approximation of the labels joint mode which (risk minimized of the $Subset\ 0/1\ loss$).

Theoretically, the product rule result is independent of the labels order. In practice, however, different chaining order may give different predictions, simply because they use different models trained on different learning sets. To reduce the impact of the chaining order, Read *et al.* propose to use a committee of chaining models, each learned on a different label order, then average, label by label, the decision of the committee predictions. But, this averaging process may also damage the consistency of the product rule approach and drift the model to minimize an undefined function that is neither the LP loss or the BR loss but some vague metric lying in between $Subset\ 0/1\ loss$ and $Hamming\ loss$.

Besides, multi-label models from the algorithm adaption category adopt generally a more direct strategy for constructing a tailored model to minimize the objective loss function [33]. For instance in [36, 37], authors propose to extends the structural SVMs to minimize the $Hamming\ loss$. Furthermore, in [39], authors introduced a generalization of SSVMs that can be implemented for optimizing a variety of multi-label loss metrics.

Moreover, algorithm adaption models are also inspired by the boosting techniques aiming to minimize the objective loss function. In [33], Amit *et al.* introduce a label covering loss function aiming to generalize the loss function optimized by the boosting strategy, that includes as special cases the $Hamming\ loss$ and the $Subset\ 0/1\ loss$.

Much more problematic, however, is the analysis of the loss function optimized by multi-label algorithms based on decision trees, *i.e.*, MLC4.5 and PCT). Indeed, the loss function optimized is more complicated since that the adaptations based on a surrogate strategy that averages the scores over the labels in the inner model construction. Thus, it is unclear what these models really manages to estimate, and what loss function they attempt to minimize. Consequently, the loss function optimized by multi-label models following similar scheme remains unknown.

2.5.4 Threshold Calibration

As mentioned in Section 2.2.1, the thresholding function is a decision function that transforms the multi-label score outputs to crisp label outputs. It is either implemented as a function learned to predict dynamically the relevant labels for each instance (*dynamic decision function*), or as a *static function*, being a constant (or a vector of constants) that draws the model decision borders between relevant labels and irrelevant labels [23].

Dynamic decision functions use a stacking-style procedure to calibrate a specific threshold for each instance [8, 28, 29, 65, 66]. The main idea behind a dynamic thresholding function $\tau(\cdot)$ is to learn a model that minimizes $|\lambda_j \in Y : s^j(x) \leq \tau(x)| + |\lambda_j \in \bar{Y} : s^j(x) \geq \tau(x)|$. In doing so, $\tau(\cdot)$ can be seen as an instance based strategy which calibrates a Single-threshold [23].

On the other hand, static decision function can calibrate either an overall threshold for all labels (Single-threshold) or a separate threshold per label (Multi-threshold). One of the simplest technique to set a Single-threshold is *RCut* [22]. For each instance's predicted scores, *RCut* considers as relevant the τ top scored labels. Thus, *RCut* is an instance based decision function that takes values in $\{0, \ldots, q\}$ and outputs a fixed number of labels. The thresholding function in *RCut* can be either specified by the user or considered as the label cardinality of the learning data set [3]. It can also be automatically tuned using a validation data set or via a Cross-validation procedure [22]. A similar label-wise Single-threshold technique is to consider a label as relevant if its associated score is greater than a calibrated fixed constant function t [15, 16]. The calibration of τ can be performed for optimizing a multi-label indicator, *e.g.*, a multi-label performance measure of interest [25] or to minimize the difference in label cardinality between the training set and the test set [17].

On the other-side, the Multi-threshold decision functions use a specific threshold for each label. Consequently, the decision function is a vector of q labels thresholds $\tau = \{\tau_1, \ldots, \tau_q\} : \tau_j \in [0, 1]$. Based on this formulation, *SCut* [22] calibrates the vector of decision borders τ_i to optimize an objective multi-label metric. The thresholds τ_i in *SCut* are tuned independently. Thus, if the objective multi-label function is label decomposable (*Hamming loss*, *Macro-F1 loss*) then a single pass from each label is sufficient, otherwise, the tuning process must reiterate until convergence (*Micro-F1 loss*, *Subset 0/1 loss*). In [25], two variations of *SCut*, named *FBR.0*

and *FBR.1*, were proposed and studied to optimize *Micro-F1* and *Macro-F1* in BR models. The idea behind FBR heuristics is to iteratively update each t_i via a greedy cyclic optimization algorithm to maximize the model performances on *Micro-F1 loss* or *Macro-F1 loss*. Another variant of Multi-threshold calibration technique named *PCut* was also proposed in [22]. Unlike *SCut*, thresholds in *PCut* take values in $\{0, 1, \ldots, N\}$, where N is the size of the test data set. Thus, *PCut* requires the existence of a complete test set and its use is limited to offline multi-label classification applications [23].

Obviously, using a static thresholding function that optimizes a specific multi-label metric bounds the decision function to a specific measure, unlike the dynamic decision function, which works autonomously. Nevertheless, a dynamic decision function remains dependent on two important factors *i)* the choice of learning model and *ii)* the input space construction (which is more complex to handle compared to simply selecting an objective function).

On the other hand, static decision function can easily lead to overfitting, especially when calibrating Multi-threshold over a validation data set [23, 24]. In [23], Ioannou *et al.* proposed a theoretical and empirical comparative study of static thresholding techniques over the *Hamming loss* as a multi-label loss function of interest. They come up with the conclusion that calibrating one Single-threshold remains the most promising technique. Moreover, the study attributes the success of the technique to the number of optimized parameters (only one threshold) which attenuate the overfitting risk. Moreover, in [24], an analysis of the optimization strategies proposed in [25] concluded that the optimization of specific performance measures on a given data set can easily lead to overfitting. Empirical results were confirmed by the theoretical study on threshold optimization for *F1* metrics [67], which demonstrates that *Micro-F1* could be optimized by predicting all instances to be negative for high imbalance labels.

2.6 Chapter summary

The study of multi-label models is an active research area, with a lot of different ensemble multi-label models being proposed in the literature. This chapter introduced the multi-label learning and reviewed the existing multi-label models. We first presented the multi-label classification terminology and defined the classification task. We also presented the different evaluation metrics used in the multi-label context. Next, we reviewed the current research on multi-label classification algorithms. We therefore discussed the optimality in the multi-label models and highlighted the challenges brought by the label dependence in the multi-label model prediction, along with their influence on the model performance. Finally, we presented the threshold calibration, an important technique that emphasizes the prediction performances and enables tailoring the prediction outputs to a specific loss metric.

From this overview, we observed that the majority of the proposed multi-label learning algorithms take foundation in classic single-label learning (*i.e.*, problem transformation models, problem adaptation). Furthermore, ensemble multi-label models have inspired several works in multi-label learning, and represent a category of classifiers that are based on top of a committee of single multi-label models, with the goal of combining their outputs as single final prediction.

In order to get the best use of the ensemble multi-label models, one needs a better understanding of the ensemble paradigm. Thus, in the next chapter, we give a description of key elements in ensemble classifiers, along with an explanation of their prominent role in enhancing the classification performances over single multi-label models. We also give an overview of the existing ensemble learning models as well.

Chapter 3

Ensemble learning

Ensemble methods, also known as committee-based models or multiple classifier systems, are a general classification system in machine learning. They build a set of base-models and combine their predictions, in contrast to ordinary single-learning models. Ensemble models were originally developed to reduce the variance in order to improve the accuracy of traditional machine learning models, ensemble models are shown to be very beneficial for enhancing the generalization ability of a single classifier, which widely influenced the development in Data Mining and Machine Learning in the last couple of decades (bagging [10], boosting [68, 69], Bayesian averaging [70], and stacking [71], to name a few).

In this chapter, we first present the committee models concept, and explain how this class of methods is broadly effective. Then, we focus on single-label learning to present the bagging and the Random Forest as both are extensively studied in the literature and relevant to this thesis. Next, we present the ensemble framework in the multi-label classification and review the state-of-the-art of ensemble multi-label algorithms.

3.1 Ensemble paradigm

Ensemble learning consists in training multiple models to jointly accomplish one common task. This category of models is based on the idea that improved performance can be achieved by consolidating the prediction of multiple models, instead of just using a single one in isolation. In the literature, such frameworks are usually named *committee* or *committee models*. The main concept in this category of models is twofold i) train a committee of individual models and ii) combine their output to deliver more accurate predictions. The principle is to give, for each base model a separate perspective of the same learning task to give more accurate predictions when consolidating their predictions.

For classification tasks, an ensemble classifier incorporates a number of sub-models called base-classifiers. Base-classifiers are usually obtained by training a base learning algorithm (Decision tree, Neural Network, k-Nearest Neighbors or other kinds of learning algorithms). Commonly, ensemble classifiers are based on the same learning model to produce a set of *homogeneous* models. Although, ensemble classifiers can also use multiple learning models to generate a *heterogeneous* committee.

The improvement of performances within the family of ensemble methods relies on the concept of diversity, which states that a good ensemble is the one in which the misclassified examples are different from one individual classifier to another. Hence, various strategies are used to obtain a group of diversified base-classifiers, whose diversities are mostly encouraged by several alternative manners, *e.g.*, sub-resampling training data, feature subsets selection, etc. Dietterich [72] explained the improvement led by ensemble models accordingly to the following three fundamental reasons:

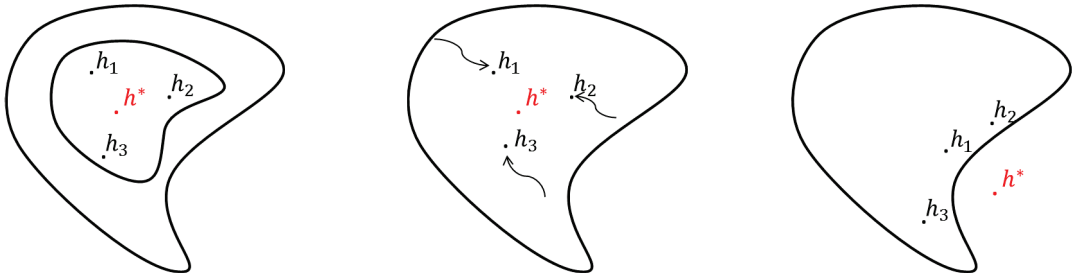
- Statistical: In general, for a given training data set, the space of potential classifiers can be too large to explore with potential classifiers sharing a similar training performance and with different unknown generalization performances. Therefore, selecting a single classifier may increase the risk of selecting a wrong classifier with a poor generalization ability. A safer option is to use all the base-classifiers and combine their outputs. Such strategy might not be better than the single best classifier h^* but will reduce the risk of choosing a wrong classifier. Dietterich gives an illustration of this argument as shown in Figure 3.1-a.

- Computational: Several learning models are based on random search or perform a local search, which causes the model to be sensitive to local optima. Even when enough data are available, finding the best hypothesis may be tough. However, running a set of different models from many different starting points may lead to different local optima, and combining all classifiers can reduce the risk of getting stuck in a local minimum. Figure 3.1-b depicts this situation.

- Representational: In many machine learning tasks, it is possible that the considered classifier space does not contain the optimal classifier. However, an ensemble of classifiers can approximate the true unknown classifier and may expand the space of representable classifier. Figure 3.1-c gives an illustration of this argument given Dietterich [72] where the optimal classifier h^* is outside the space of considered of classifiers.

Ensemble models work in two steps a) *The training step* and b) *The prediction step*. Figure 3.2 shows a common ensemble classifier architecture. *The training step* aims to generate a committee of base-models from a training data set using a base-learner generator. In the training step, we can distinguish two main architectures of ensemble models:

FIGURE 3.1: Fundamental reasons for combining base-model predictions: the outer curve represent the space of all possible models. h^* is the true model for the problem, and h_i's are learned base-models. The inner curve in (a) is the space of models with the same performances on the training data. Graphical illustrations based on similar figure in [72].



(a) Statistical *(b) Computational* *(c) Representational*

- **Parallel**: In this architecture, the base-classifiers are trained in parallel and independently from each other as depicted in Figure 3.2-(a)). It is the simplest and the most popular ensemble architecture as it has the advantage of being easy to use and can be implemented in parallel.

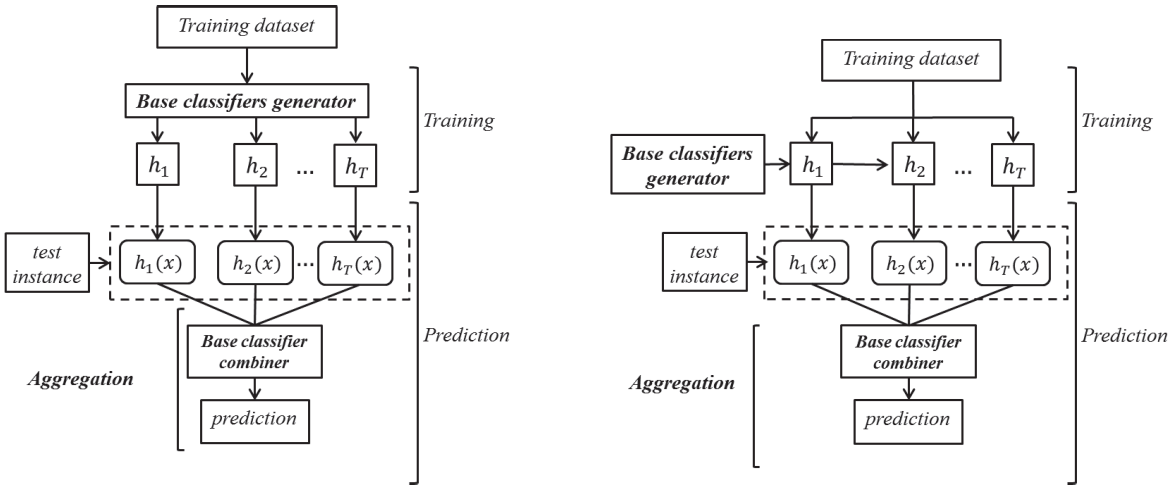
- **Serial**: In this architecture the base-classifiers are trained sequentially (illustration given in Figure 3.2-(b)). This variant of ensemble models involves an iterative training where a specific error function is used to train each base-model depending on the performance of the previous ones. Ensemble models based on this architecture are known as boosting models.

Besides, *the prediction step* intends to label unseen instances with the ultimate purpose of merging the committee predictions. *The prediction step* works in two phases: first, each individual model provides its prediction outputs, second, all predictions are combined to form the final ensemble prediction. The combination step consists in an aggregation scheme of the predictions (typically through simple or weighted averaging) based either on the crisp class predictions or the probability predictions. The appropriate combination scheme depends on the type of information obtained from each individual model and also on the output information expected from the ensemble model.

3.2 Committee construction

As aforementioned, the principle of this category of algorithms is to train different base models with the idea to come up with a final prediction that is the combination of the predictions given by each individual model. The simplest approach for combining committee predictions is to average the predictions for each new instance. From a frequentist perspective, this is motivated by the trade-off between bias and variance, which decomposes the prediction error of a model into the

FIGURE 3.2: Emsemble models architecture



(a) Parallel training *(b) Serial training*

bias component arising from the difference between the trained model and the true function to estimate, and the variance component that expresses the model sensitivity to individual data samples.

In practice, only one single data set is available, so it is necessary to come up with some way to introduce variability among the committee. One strategy is to use the bootstrap data sets. Before introducing the bootstrap strategy, let assume there is a model class $\mathcal{H}^w$ where we generate the base-classifiers $h \in \mathcal{H}^w$. Let H denote the ensemble framework combining multiple base models and let t index the t^{th} base-classifier. In this section, let consider a mono-label classification problem in which we aim to predict the value of a multi-class target where $B = \{(\mathbf{x}_{(1)}, y_{(1)}), (\mathbf{x}_{(2)}, y_{(2)}), \cdots, (\mathbf{x}_{(n)}, y_{(n)})\}$ represents its associated training data set.

3.2.1 Bootstrap Aggregation

Also known as Bagging [10], it consists on a vote different classifiers generated by different bootstrap samples [73]. A collection of T bootstrap samples, B_t, with $t = 1, \cdots, T$, are generated from the training data. The bootstrap data sets are used to train separate copies of the base-classifier h_t. Each bootstrap is generated by uniformly sampling with replacement n instances from the training data set. The final committee classifier H is built from $h_1, \cdots, h_T$ and outputs the class predicted most often by the committee, with ties broken arbitrarily. The bagging prediction is defined by :

$$H(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \frac{1}{T} \sum_{t=1}^{T} \mathcal{I}(h_t(\mathbf{x}) = y). \tag{3.1}$$

This combination scheme can be seen as a simple averaging process overall the base-classifiers in the committee aiming to reduce the variance. Indeed, the Bagging exploits the independence between base-classifiers to give more accurate predictions [1]. This is based on the fact that errors can be dramatically reduced by combining independent base models. In fact, the Bagging gives an incorrect prediction when at least half of the base-models make incorrect predictions. Assuming that each base-classifier has a probability ε to produce an independent miss-classification: $p(h_t(\mathbf{x}) \neq y) = \varepsilon$, the probability of the Bagging making an incorrect prediction is given by:

$$p(H(\mathbf{x}) \neq y) = \sum_{t=0}^{T/2} \binom{T}{t} (1-\varepsilon)^t \varepsilon^{T-t} \leq exp\left(-\frac{1}{2}T(1-\varepsilon)^2\right). \tag{3.2}$$

The probability decreases exponentially with the number of base-classifiers and approaches zero when the committee size approaches infinity. This result suggests that the average error of a model can be reduced by a factor of T simply by averaging T versions of the model. However, it depends on the assumption that the errors due to each model are uncorrelated. The purpose of the Bootstrap sampling is to best exploit the independence by adding perturbation to enhance diversity within the committee. Indeed, the bagging consists in estimating the $E_p h(x)$ where each $(x, y) \sim p$. Thus, the bagging formulation (3.1) can be seen as a Monte Carlo estimate of the model prediction, approaching it as the committee size approaches infinity. In other words, $H(x) \to h(x)$ as $T \to \infty$ [1]. Thus, the bootstrapping is considered as a way of assessing the accuracy of a prediction. In practice, the improvement also depends on the base learner used to learn the base-classifiers. The performance improvement is important if the base learner is unstable (*e.g.*, decision trees) and the induced models are good and not correlated. While, bagging stable algorithms (*e.g.*, k-nearest neighbor) may not lead to good performances [10].

It is important to note that, when the bootstrap samples are generated from the data, they seem to be similar. However, they are not identical since that each bootstrap will cover only around 63% of the initial training data set under the condition of a large data set. Given a training set of n instances, each bootstrap is a subset of size n generated by sampling with replacement n times from the original training data. Thus, some observations do not appear in the bootstrap sample.

The probability that the i^{th} training instance is not sampled once is $(1 - 1/n)$, and the probability that it is not sampled at all is $(1 - 1/n)^n$. For large n, this probability approach $\frac{1}{e} \simeq 37\%$. In other words, each bootstrap sample contains only about 63% of unique instances, meanwhile 37% of instances will not appear in the bootstrap. These later instances are called Out-of-bag (*Oob*) samples. They provide an effective way to estimate the generalization error of the base learner known as out-of-bag estimation [74–76].

[1] Note that the bagged estimate will differ from the original estimate when the latter is a nonlinear or adaptive function of the $\mathbf{x}$.

3.2.1.1 Bagging to estimate probabilities

Frequently,the class-probability estimates (at $\mathbf{x}$) is required rather than a direct classification. In such case, it is tempting to consider the voting proportions $S^y(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} \mathcal{I}(h_t(\mathbf{x}) = y)$ as an estimation of these probabilities. A simple binary classification example confirms that they fail in this regard. Suppose the true probability of class $y = 1$ for a given x is 0.75, and each of the bagged classifiers models predict accurately 1. Then $S^{y=1}(\mathbf{x}) = 1$, which is incorrect. For many base-classifiers there is already an inner function that estimates the class probabilities at a given $\mathbf{x}$. For instance, the estimation of the class probability in a decision tree is the class proportion in the terminal node. In such case, the decision process of the Bagging can be softened by considering the probability outputs of the base-classifiers, instead of the crisp prediction. As long as each of the models gives posterior probabilities for the classes, it is possible to combine the outputs systematically using the average of their probabilities predictions, *i.e.*, $S^y(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T}(s_t^y(\mathbf{x}) = y) : y \in \mathcal{Y}$, where the finale committee class prediction is as follows:

$$H(\mathbf{x}) = \underset{y\in\mathcal{Y}}{\operatorname{argmax}} \frac{1}{T}\sum_{t=1}^{T}(s_t^y(\mathbf{x}) = y).$$

Furthermore, the bagged committee can also give a probabilistic interpretation to the model outputs in order to provide a fully probabilistic mixture of models (Bayesian model averaging) with an accurate estimation for the $p(y|\mathbf{x})$. This strategy does not only produce improved estimates of the class probabilities, but also manages to provide bagged classifiers model with lower variance, especially for small T [1].

The bootstrap method provides a straight computational way of assessing uncertainty, by only sampling from the training data. It is also important to highlight that no other information about $h(\mathbf{x})$ is required in the combination step except that each base-classifier takes the input vector $\mathbf{x}$ as a parameter and generate an output $y \in \mathcal{Y}$. The bootstrap method is "model-free," since it is based only on the instance samples, not a particular parametric model, in order to generate the bootstraps. To achieve more significant improvements, more sophisticated committee construction techniques are proposed such as *Random Forest*.

3.2.2 Random Forest

As suggested by the name, a Random forest (RF) [11] is a tree-based ensemble model were each tree is depending on a set of descriptive features. It is an extension of the bagging with more randomness on the inner decision tree predictors to obtain more diverse classifiers. The main idea is to use a collection of unpruned decision trees (unstable models) as base classifiers and introduces additional randomness into all trees. Namely, in each interior node of each tree, a

subset of r inputs variables are randomly selected and evaluated with the Gini index heuristics. The variable with the highest Gini index is chosen as a split in that node. The number of the selected features r, also known as *mtry* parameter, is usually fixed to $\sqrt{f}$, or $log(2 \times f)$ where f is the dimension of the input feature space.

Random Forests can be used for either a categorical target variable or a continuous target variable. Similarly, It can also handle both continuous and categorical input features. From a computational standpoint, Random Forests are a popular machine learning model since they are relatively fast to train and easy to use in the prediction. It also has the advantage of having a reduced number of parameters and can easily be implemented in parallel. Furthermore, Random Forests are appealing because of the additional possibilities they provide, such as feature importance measure, a built-in estimate of the generalization error and missing value imputation [12].

3.2.3 Using Out-of-Bag samples for error estimation

As aforesaid, when a bootstrap sample is generated from the data set, some observation does not appear in the bootstrap. These "out-of-bag" samples are extremely useful for estimating the generalization error and the variable importance in the RF model.

To estimate the generalization error of the model, one cannot use observations that were in the training data, and have to use only data that has been outside the training set. The alternative idea in the bootstrap models is to take advantage from the out-of-bag instance as they were not used as training samples in the base models. For this reason, the predictions for observations that were in the original training set are only performed using the base-classifiers where these observations were out-of-bag, where these predictions are known as out-of-bag predictions. For classification, the generalization for the 0/1 loss error rate is estimated using the out-of-bag is given by:

$$\mathcal{E}_{Oob} = \frac{1}{n} \sum_{i=1}^{n} \mathcal{I} \left(H_{Oob}(\mathbf{x}_{(i)} \neq y_{(i)} \right). \tag{3.3}$$

It is important to highlight that the out-of-bag error rate is not obtained by computing the out-of-bag error rate separately for each individual base-classifier to be averaged over the committee. Instead, it is computed using the error rate of the out-of-bag predictions. Algorithm 1 details the out-of-bag predictions process.

Algorithm 1 *Out-of-Bag Predictions*

Require:

The training data set $B = \{(\mathbf{x}_{(1)}, y_{(1)}) \cdots (\mathbf{x}_{(n)}, y_{(n)})\}$;

The set of bootstrap samples $\{B_1 \dots B_T\}$;

The committee of base classifiers $\{h_1 \dots h_T\}$;

1: **for** $i \in \{1, \dots, n\}$ **do**

2: $\quad \mathcal{J}_i \leftarrow \{t : (\mathbf{x}_{(i)}, y_{(i)}) \notin D_t\}$

3: $\quad J_i \leftarrow$ cardinality of $\mathcal{J}_i$

4: $\quad H_{Oob}(\mathbf{x}_i) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \frac{1}{J_i} \sum_{j \in \mathcal{J}_i} \mathcal{I}\left(\hat{h}_j(\mathbf{x}_{(i)} = y\right)$.

5: **end for**

3.3 Ensemble Multi-label models

As presented in the previous section, ensemble approaches are proposed in traditional mono-label learning to improve the robustness and the predictive performance of a weak classifier. On the other hand, in multi-label classification tasks, ensemble models have been suggested for the same reasons and also to overcome other issues that are specific to the multi-label setting (the computational complexity of LP approach [15] or the independence assumption of BR models [17]). In this context, ensemble multi-label models are defined as meta-algorithms based on the top of common multi-label learners [9].

In this section, we give an overview of the state-of-the-art ensemble multi-label models. We follow the same categorization proposed by Tsoumakas and Katakis [3] and distinguish two main categories of ensemble multi-label models: *a) Ensemble models based on adaptation methods* and *b) Ensemble models based on transformation methods*.

3.3.1 Ensemble models based on adaptation methods

Algorithm adaptation based ensembles consist of base-classifiers that are adaptation multi-label algorithms [9] (see Section 2.4.1). They are based in the top of extended and tailored machine learning algorithm for the multi-label task, *e.g.*, decision trees [6, 32], and k-nearest neighbors [77].

3.3.1.1 Random Forest Predictive Clustering Tree (RFPCT)

Kocev *et al.* [32] presented a Random Forest multi-label ensemble model named *Random Forest Predictive Clustering Tree* RFPCT (see Section 2.4.1). The RFPCT approach is based on the

top of the *Predictive Clustering Tree* algorithm [78]. The diversity in the ensemble committee is carried out by the bagging strategy along with a random subset selection of the input features at each node of PCT as in the Random Forest model [11]. During the prediction step, each base-model outputs its multi-label predictions, which are then combined via a label voting scheme, *i.e.*, using typically a majority or a probability distribution vote for each label separately.

3.3.1.2 Random Forest of Multi-Label-C4.5

Another version of multi-label Random Forest based on the top of the ML-C4.5 [6] was proposed in [9] and named *Random Forest of ML-C4.5* (RFML-C4.5). The ensemble model follows the same construction philosophy as in RFPCT: the diversity is carried out using the bagging strategy and a random selection of a subset of variables in each tree node. The used ML-C4.5 is an adaptation of the well-known C4.5 algorithm to the multi-label setting, where the definition of entropy is modified to allow multiple labels in the leaves (see Section 2.4.1). In the prediction step, the RFML-C4.5's base-classifiers are combined using either a crisp label or probabilistic vote over each label.

3.3.1.3 Variable Pairwise Constraint projection for Multi-label Ensemble (VPCME)

Recently, a novel multi-label classification framework called *Variable Pairwise Constraint projection for Multi-label Ensemble* (VPCME) [77] was proposed. The framework extends the traditional pairwise constraints projection to the multi-label task. The diversity within the base-classifier committee is carried out by re-sampling the pairwise constraints to learn, for each base-classifier, a different lower-dimensional representation of the input space that preserves the correlations between samples and labels. After that, the base-classifiers are learned using boosting-like strategy in order to improve the generalization ability of each committee member. VPCME is different from other adaptation ensemble multi-label models, in the sense that it offers the possibility to use any multi-label classifier and adapts the boosting to the multi-label context.

3.3.2 Ensemble models based on transformation methods

3.3.2.1 Ensemble of Binary Relevance classifiers (EBR)

The most simple multi-label model is the EBR classifier which is based on the top of the popular multi-label *Binary Relevance* classifier (BR). In its original version in [17], each base-classifier in EBR is carried out on a random sub-sampling of the training data set. For the multi-label classification of a new instance $\mathbf{x}$, each base-classifier h_t provides its binary predictions $h_t^i(\mathbf{x})$

for each label λ_i. Subsequently, the EBR calculates the average decision for each label λ_i and outputs a final positive prediction if the average prediction for a label is greater than 0.5.

3.3.2.2 Ensemble Classifier Chain (ECC)

To tackle the chain order in CC, Read *et al.* [17] proposed the *Ensemble of Classifier Chains model* (ECC). Indeed, as the order of the chain can influence the CC performance, the idea in ECC is to train a committee of CC models, each based on random chain orderings, and on a random subset of training instances. In the prediction step, ECC combines the base-classifiers outputs via label vote, where a label is assigned to an instance if predicted accordingly by the majority of base-classifiers, *i.e.*, if the average prediction for a label is greater than 0.5.

3.3.2.3 Ensemble of Pruned Sets (EPS)

Similarly to the EBR models, the strain forward ensemble multi-label model based on the LP is the ELP model. The diversity within the committee is conducted using a bagging strategy. Inspired by this simple strategy, researchers proposed sophisticated base-classifiers in order to bypass the LP complexity drawbacks. In fact, Read *et al.* [16] proposed the *Ensemble of Pruned Sets* (EPS). First, the model deals with the LP complexity and prune samples with rare labelsets to let the model focus on the most important ones. Then, the model compensates the information loss by reintroducing the pruned sample associated with the frequent subset of their original labelsets. It is noteworthy that an LP model is not able to output labelsets that are not in the training set. EPS trains a committee of LP classifiers, each trained on a random selection of samples. Furthermore, during the prediction stage, EPS specifically uses a label voting scheme, where a majority threshold separates relevant labels to expand the generalization of the model by predicting additional labelsets being outside the training set [16].

3.3.2.4 Ensemble of RAndom k-labELsets (RAkEL)

To keep LP's advantage (modeling the joint distribution) while overcoming its considerable shortcomings, Tsoumakas *et al.* proposed an effective and more popular ensemble method named *RAndom k-labELsets* (RAkEL) [15]. The idea behind RAkEL is not only to construct a committee of base-classifiers to enhance the quality of the single model, but also to trade off the BR label independence assumption with LP complexity. The main innovation introduced by RAkEL in the realm of ensemble multi-label models, is the way in which the diversity is promoted within the committee. Indeed, this diversity is established in the target space rather than the feature space as in traditional ensemble models. Several other works have been inspired by this idea and proposed extensions of the RAkEL algorithm [79–81].

Each base-classifier in RAkEL is an LP multi-label model, specialized on a small random subset of k labels (k-labelsets). By construction, RAkEL takes into account the correlation between the labels within the same k-labelsets, and at the same time, reduces the number of labels handled by each LP.

For each base classifier h_t, the algorithm selects (randomly and without replacement) a k-labelsets from all distinct subsets of k sized labels. The number of all possible k-labelsets is given by $\binom{q}{k}$. Then, it learns an LP base classifier $h_t : \mathcal{X} \rightarrow \mathcal{L}_t^k$ to learn to predict the label appearing on its own k-labelsets. The prediction of a new instance is achieved by combining the committee crisp labels outputs through a label vote by considering all the base-classifiers.

In the prediction step, each base classifier provides a binary decision $h_t^i(\mathbf{x})$ for each label λ_i in its corresponding k-labelsets k-$\mathcal{L}_t$. Subsequently, RAkEL computes the average decision separately for each label λ_j in $\mathcal{L}$. The final committee decision for λ_j is positive if the average is greater than 0.5, otherwise the instance is not associated with the label. A formal description of the RAkEL model is given in Algorithm 2. Despite its intuitive appeal and competitive performance, RAkEL suffers a lack of theoretical understanding. For instance, it is not clear what loss function it intends to minimize.

As the first extension of RAkEL, Kouzani *et al.* combine a random selection of labels, a random feature subset, and a random instance subsets, to build a Triple-Random Ensemble Multi-Label Classification (TREMLC) [79]. Each base-classifier in TREMLC is trained using a portion of data (drawn randomly without replacement) and trained to predict k-labelsets using only a subset of features. The authors reported that the model performance was especially susceptible to the percentage of instance selection and the random subspace size. In fact, such diversity is hard to manage and requires a large ensemble size. However, the ensemble size depends on the number of labels since the k-labelsets selection is carried by a random selection without replacement from all possible k-labelsets in $\mathcal{L}$. In [80], an improved version of RAkEL named RAkEL++ is presented [15]. The idea is to, *i)* aggregate the probabilities provided by the base-classifiers rather than using the 0/1 votes as in the original RAkEL, and *ii)* use a single threshold for all labels, calibrated by optimizing a performance measure of interest via a cross-validation (CV) procedure.

3.3.3 Other ensemble multi-label methods

Other than the aforementioned ensemble methods, some multi-label approaches are occasionally referred to as ensemble methods, in the sense that they involve multiple classifiers. This include the well known HOMER algorithm by Tsoumakas *et al.* in [54], *Pair-wise methods* such as Calibrated label ranking (CLR) [82] and QWeighted approach to multi-label learning (QWML) [47]. Also, other models extended the ensemble paradigm to handle the multi-label tasks, rather

Algorithm 2 RAkEL: Ensemble of RAndom k-labELsets

Require: Training multi-label data (D); Set of labels ($\mathcal{L}$); k-labelsets size (k); Ensemble size (T).

B- Training

$H \leftarrow \emptyset$
$R \leftarrow \mathcal{L}^k$
for $t = 1 : T$ **do**
  $\mathcal{L}_t^k \leftarrow$ randomly select a k-labelsets from R
  train an LP classifier $h_t : \mathcal{X} \rightarrow \mathcal{L}_t^k$ on D
  $R \leftarrow R \backslash \mathcal{L}_t^k$
  $H \leftarrow H \cup h_t$
end for

B- Prediction

Require: Test instance **x**

$Sum \leftarrow 0; Votes \leftarrow 0$
for $t = 1 : T$ **do**
  for $\lambda_j \in \mathcal{L}_t^k$ **do**
    $Sum_j \leftarrow Sum_j + h_t^j(\mathbf{x})$
    $Votes_j \leftarrow Votes_j + 1$
  end for
end for
for $j = 1 : q$ **do**
  $Avg_j \leftarrow Sum_j / Votes_j$
  if $Avg_j > 0.5$ **then**
    $H^j(\mathbf{x}) \leftarrow 1$
  else
    $H^j(\mathbf{x}) \leftarrow 0$
  end if
end for

than the classification algorithm itself. This includes ADABOOST.MH and ADABOOST.MR [83] and their variants [84, 85], which are two extensions of the well-known ADABOOST on multi-label data. The aim of AdaBoost.MH is to minimize the Hamming loss, meanwhile ADABOOST.MR is designed to minimize the ranking loss. During the training phase, the original multi-label task is transformed into a binary problem, and a set of weights are maintained for both instances and labels through the iterating process.

In this thesis, these methods are not considered as ensemble multi-label models, since that the multi-label problem is decomposed into one binary mono-label task managed with an ensemble model, *i.e.*, the inner base-models are not multi-label models. In contrast, an ensemble multi-label model directly manages multi-labeled data using a committee of multi-label models [2, 9]. Thus, these models are considered beyond the scope of this thesis.

Besides, an other group of heterogeneous models distinguishes itself. This type of committee-based models aims to use different multi-label learners as base-classifiers to improve the global committee performance. The diversity in heterogeneous ensemble multi-label models is carried out not only by classical instance-based diversity, *i.e.*, random instance selection or *bagging*, but also by the dissimilarity of the base-learner [86].

3.4 Chapter summary

In this chapter, we presented the ensemble learning paradigm. We first presented the main idea behind this category of models and showed how they enhance the generalization performance of a single classifier. Then, we presented in more details the two key components of these models, *i.e.*, the *committee generation* and the *base-classifier aggregation*. Next, the chapter presented the bootstrap aggregation as the classical ensemble models since it is closely related to our contribution in this thesis.

Finally, the chapter introduced the ensemble multi-label models and gave an overview of the recently proposed algorithms and discussed their strategies in the light of the two main categories of multi-label models: Algorithm adaptation approaches and Problem transformation approaches.

However, we noticed that most works in ensemble multi-label paradigm often propose a new ensemble model (based on a new committee construction strategy) while lacking a rigorous analysis of the combination step and its consistency with the committee construction. In the next chapter, we investigate the consistence between the committee generation and the base-classifier aggregation in ensemble k-labelsets models (RAkEL). Furthermore, we highlight the importance of the combination step over the model performance and suggest a new committee construction together with an adequate committee combination to enhance the prediction quality.

Chapter 4

Calibrated k-labelsets for Ensemble Multi-Label Classification

Ensemble multi-label k-labelsets models are efficient and computationally practical approaches. Their greatest concern in breaking down the multi-label tasks in a set of smaller ones where the links between the labels can be modeled easily. The idea behind these models is to train a committee of multi-label models each specialized in a smaller multi-label set. RAndom k-labELsets (RAkEL) is the most popular k-labelsets ensemble multi-label approach. Each base-model in RAkEL is a LP model trained on a small random subset of k labels. Unlike traditional ensemble, where the diversity within the base-model is created in the input space, the diversity in this category of multi-label models is basically carried out in the output space. By construction, the model aims to consider the label structure within each base-model, and at the same time, reduce the number of labels handled by each LP. In the prediction step, the labels associated with a new instance are given by the aggregation through a label majority voting process of the binary outputs of each base-model in the committee.

This Chapter, examine the RAkEL model as the basic k-labelsets multi-label approach and point out some weaknesses within the model committee construction raised by the imbalanced label representation. Then, we propose three practical solutions to overcome these drawbacks in a new Calibrated k-labelsets committee [81].

4.1 Committee construction in the RAkEL model

As described in the Section 3.3.2.4, the diversity in ensemble k-labelsets models is carried out in the output space by a random selection of T k-labelsets ($\mathcal{L}^k$) without replacement from the set of all possible label sets of size k in $\mathcal{L}$.

This committee construction allows the base classifier having different parts of the multi-label classification task while sharing some target labels (i.e., labels appearing simultaneously in several selected k-labelsets). The overlapping character of the k-labelsets selection allows the committee to gather multiple predictions for the same label by the different base-models. Furthermore, as the different base-models are trained on different label spaces, it offers a diverse perspective for each label prediction considering that, in each k-labelset, a label appears with a different subgroup of labels. Thus, combining the predictions made by the committee of the base-classifiers through a voting process, offer the possibility to correct potential uncorrelated errors and improves the overall performance. To guarantee the effectiveness of this reasoning, it is necessary to ensure that each k-labelsets does not appear more than once within the committee; as this may damage the voting procedure. For this purpose, the random k-labelsets selection in RAkEL is conducted *without replacement* from the set of all possible label sets of size k in $\mathcal{L}$ and not by randomly selecting subsets of labels of size k from $\mathcal{L}$.

However, as all heuristic methods, RAkEL has several shortcomings:

- First; during the k-labelsets sampling process, some labels are selected less often than others, hence creating an imbalance label representation within the selected labelsets (i.e., some labels are over selected meanwhile others are rarely selected (or never selected)). Obviously, the probability predicted for a label appearing in several k-labelsets is more accurate than the probability predicted for label appearing in one k-labelset. However, aggregating naively the committee predictions regardless of this imbalance may reveal inconsistency over the confidence of each label probability estimates. The following example illustrates the potential problem with such aggregation. Assume that λ_1 and λ_2 appear respectively in 10 and 3 k-labelsets. If for a test sample ($\mathbf{x}_{(t)}$), 9 base-classifiers predict λ_1 and 3 classifiers predict λ_2, the probabilities to assign λ_1 and λ_2 to $\mathbf{x}_{(t)}$ given by the original RAkEL are respectively $Sl^1(\mathbf{x}_{(t)}) = (9/10) = 0.9$ and $Sl^2(\mathbf{x}_{(t)}) = (3/3) = 1$. The confidence in the probability prediction based on 3 classifiers is not as good as for 10 classifiers of course.

  Furthermore, the question remains on how the model should predict a label which has never appeared in the committee? In fact, this highlights the shortcoming of the model to adequately i) control the k-labelsets generation ii) cover the label space and iii) to balance the label representation in the committee.

- Second; considered as an ensemble approach, RAkEL fails taking advantage of the best part of the diversity concept when constructing its base-classifiers, since that, each label combination is allowed to appear at most once. Even if this choice is motivated by the fact that prediction error should be uncorrelated to be reduced, nothing hinders two base-classifiers to share the same output space if the diversity is maintained in the input space. On the contrary, this may improve the predictive performance of the ensemble since more

votes could be performed for each k-labelsets leading to more accurate estimates of the true value of the k-labelsets.

- Third; the use of a unique 0.5 threshold to select the final predicted bi-partition is not managed in accordance with the label imbalance representation within the committee and also does not suit data sets where labels are associated with few training examples which is the case of the multi-label classification task [17].

4.2 CkMLC: A New k-labelsets ensemble model

In this section, we discuss a Calibrated k-labelsets for Ensemble Multi-Label Classification method (termed CkMLC as a shorthand) to improve the overall performance of k-labelsets based ensemble models. Our contribution is three-fold: First, we use Bagging in tandem with random k-labelsets to increase the diversity of the base classifiers and thus the robustness of the ensemble. Second, the label set probabilities are calibrated to account for the effective label occurrence rate in the random labelsets sampling. Third, a finely-tuned threshold is associated to each label instead of using a single threshold for all the labels [15].

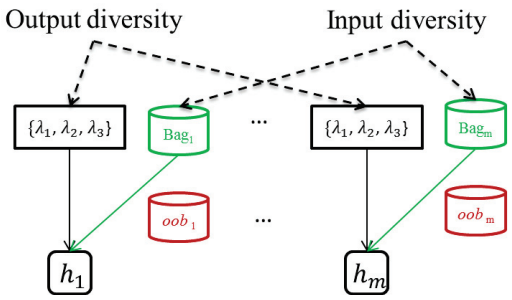
4.2.1 Committee construction

To fully benefit from the ensemble paradigm, we propose to expand the committee size by increasing the diversity within the base-classifiers. From this perspective, we propose to induce diversity also in the input space by a bootstrapping strategy. The latter will allow multiple base-models sharing the same target space while preserving the input diversity. In contrast to [79] we enforce diversity only in instance space via random sampling with replacement from the instance set. Combining the input and the output diversity has two advantages: The k-labelsets strategy provides a specific output view to the base-classifier. Meanwhile, the latter strategy enforces diversity by allocating distinct samples to the classifiers. Last but not least, in each bootstrap, almost 33% are left out-of-bag (Oob), *i.e.*, they are not used for the construction of their corresponding model. These samples can be used as an unbiased validation set for the threshold calibration. Figure 4.1 shows the CkMLC architecture.

4.2.2 Adaptive base-model combination

As illustrated above, the committee construction in ensemble k-labelsets induces different predictions for each label. To cope the imbalance in the labels representation, we aim to consider

FIGURE 4.1: Calibrated k-labelsets Multi-Label Classifier committee construction



the number of the base-classifiers used for the prediction of each label. Therefore, we propose to smooth the ensemble probability estimate for each label using the *Laplace estimate* as:

$$Sl^i(\mathbf{x}) = \frac{\left(\sum_{\{h_t \in H | \lambda_i \in \text{k-}\mathcal{L}_t\}} h_t^i(\mathbf{x})\right) + 1}{|\{h_t \in H | \lambda_i \in \text{k-}\mathcal{L}_t\}| + C} \tag{4.1}$$

Where C is the number of classes per label, in our case $C = 2$. In the previous example, the *Laplace estimate* yields a probability of $\frac{9+1}{10+2} = 0.83$ for λ_1 and $\frac{3+1}{3+2} = 0.8$ for λ_2.

This smoothing strategy flattens the label probability distribution and improves the multi-label model performance regarding the probability-based ranking measure. It is important to note that the smoothing does not change the probability distribution regarding 0.5. *i.e.* if a probability is greater (lower) than 0.5, it will remain greater (lower) than 0.5 for the Laplace estimate.

4.2.3 Threshold calibration

To refine the scope of the CkMLC predictions, we propose to use a specific threshold for each label that considers the imbalance in the data set. We propose a simple forward algorithm easy to implement with a low computational cost for calibrating label the decision thresholds. The idea is to take advantage of the committee structure in the CkMLC and benefits from the model *Oob* instances. Thus, the calibration does not need to carry a cross-validation procedure to create a validation data set.

To select the most promising multi (separate) thresholds over a specific multi-label performance measure of interest in our *Forward Multi-label Thresholds Calibration* strategy, the best thresholds are firstly selected independently for each label $\lambda \in \mathcal{L}$. Then, the label achieving the best performance λ^* is selected as well as its optimal threshold τ_{λ^*}. Then, λ^* is removed from the search space $\mathcal{L}$ and added to $\mathcal{L}^*$. Afterward, for each label in $\mathcal{L}$ the best thresholds are selected as having the best performance jointly with labels in $\mathcal{L}^*$ associated with their calibrated thresholds. The process is repeated until calibrating all thresholds. Algorithm 3 gives a formal description of the procedure. To the best of our knowledge, this is the first attempt to propose an algorithm

for selecting a distinct threshold per label based on oob instance and without being metric dependent. The proposed thresholding algorithm is valid for all bi-partition-based metrics, including both instance-wise and label-wise measures.

Algorithm 3 *Forward Multi-label Thresholds Calibration*

Require:

Oob predictions probabilities ($\hat{Y}$); *Oob* real labels (Y); label set $\mathcal{L}$; multi-label loss metric to minimize ($MLloss$);

1: $\mathcal{L}^* \leftarrow \emptyset$; $\tau^* \leftarrow \emptyset$
2: **while** $\mathcal{L} \neq \emptyset$ **do**
3: $\quad \lambda^*, \tau^*_{\lambda^*} \leftarrow \underset{\lambda \in \mathcal{L}, \tau \in [0,1]}{\operatorname{argmin}} \; MLloss([\hat{Y}_{\mathcal{L}^*}/\tau \cup \{\hat{Y}_\lambda/\tau^*\}], [Y_{\mathcal{L}^*} \cup \{Y\}])$
4: $\quad \mathcal{L}^* \leftarrow \mathcal{L}^* \cup \lambda^*$
5: $\quad \tau^* \leftarrow \tau^* \cup \tau_{\lambda^*}$
6: $\quad \mathcal{L} \leftarrow \mathcal{L} \backslash \lambda^*$
7: **end while**
8: **return** τ^*

Most state-of-the-art thresholding strategies propose a multi-threshold calibration via a cross-validation procedure. However, the CV procedure leads to a critical issue on how to select the most promising threshold vector τ. *i)* Should the algorithm select τ as the combination of the best performing thresholds per label which should be crucial for label-wise performance metrics but not for instance-wise ones, *ii)* should the algorithm select the most promising threshold vector τ based on the performances of all possible threshold combinations? In that case, for 9 different threshold values per label ranging for example from 0.1 to 0.9 in 0.1, the calibrating threshold via CV is too intensive since it will need to evaluate the performances of 9^q threshold vectors τ.

Unlike these algorithms, our approach avoids these issues by using the out-of-bag data set; which also reduces the learning complexity since only one single model is learned and can exploit the entire training data set.

4.3 Experimental evaluation

This section investigates the effectiveness of our proposed CkMLC algorithm and show experimental studies on a broad range of real-life multi-label data sets. We first give a short description of the multi-label data sets and performance metrics used in this study. Next, we present the evaluation protocol and the parameter instantiations for the compared multi-label learning methods.

4.3.1 Data sets

To thoroughly evaluate the performance our algorithm, a variety of real-word multi-label data sets from the *Mulan's repository* [87] are employed in this section. We selected these data sets as they have already been used in various empirical studies and cover different application domains, including text categorization (Yahoo data, Enron, Medical), Image classification (Scene), bioinformatics (Yeast), music and audio classification (Emotions and Birds). In summary, 20 data sets were used with labels ranging from 5 to 53 labels and a number of examples from 194 to over 5000. Table 4.1 summarizes their basic statistics: **N** the number of examples, **M** the number of features, **q** the number of labels; ***Card*** the Label Cardinality and **LD** the Label Density (Section 2.1)

TABLE 4.1: Description of the multi-label data sets used in the experiments.

Data	**Domain**	**N**	**M**	**q**	***Card***	**LD**
Arts	Yahoo-Text	5000	462	26	1.636	0.063
Birds	Audio	645	260	19	1.014	0.053
Business	Yahoo-Text	5000	438	30	1.588	0.053
Computers	Yahoo-Text	5000	681	33	1.508	0.046
Education	Yahoo-Text	5000	550	33	1.460	0.044
Emotions	Music	593	72	6	1.869	0.311
Enron	Text	1702	1001	53	3.378	0.064
Entertainment	Yahoo-Text	5000	640	21	1.420	0.068
Flags	Image	194	19	7	3.392	0.485
Health	Yahoo-Text	5000	612	32	1.662	0.052
Image	Image	2000	249	5	1.236	0.247
Medical	Text	978	1449	45	1.245	0.028
Recreation	Yahoo-Text	5000	606	22	1.423	0.065
Reference	Yahoo-Text	5000	793	33	1.169	0.035
Scene	Image	2407	294	6	1.074	0.179
Science	Yahoo-Text	5000	743	40	1.540	0.036
Slashdot	Text	3782	1079	22	1.180	0.041
Social	Yahoo-Text	5000	1047	39	1.283	0.033
Society	Yahoo-Text	5000	636	27	1.692	0.063
Yeast	Biology	2417	103	14	4.237	0.303

4.3.2 Evaluation protocol

As the CkMLC can output either a probability score for each label or a bi-partition of the label space into crisp labels, the performance analysis will cover both type of outputs. In the sequel,

both the new ensemble construction and the threshold calibration strategies combined together in our CkMLC approach are firstly studied and compared according to score based metrics. Then, the algorithm's performances of CkMLC were analyzed over bi-partition-based metrics. CkMLC is compared with several state-of-the-art multi-label classification methods, namely RAkEL taken as our gold standard k-labelsets approach, RAkEL++ [80] and TREMLC [79] that should be viewed as another variants of RAkEL, the multi)-label classification approach FBR [25] which implement (as in our CkMLC and RAkEL++) a different thresholding strategy for the prediction step. Details about the algorithm are given in Section 2.5.4. CkMLC is also compared against EBR, ELP and ECC to assess its performances against traditional multi-label ensemble models.

Finally, the experiments cover a large group of multi-label performance measures including *Ranking loss* and *One error* to evaluate the quality of label score predictions; and *Subset 0/1 loss*, *Jaccard loss*, *Micro-F1 loss*, *Macro-F1 loss*, *Instance-F1 loss* and *Hamming loss* as metrics to evaluate the crisp labels outputs. Note that the threshold calibration should not affect probability-based metrics. However, the calibration should significantly affect the model performances over bi-partition-based metrics. A detailed description of these multi-label metrics is given in Section 2.3.

4.3.3 Experimental setup

To make fair comparisons, the parameters of each algorithm were set as suggested in the literature for yielding the most satisfactory performances. The same experimental setting in [79] was adopted here for the RAkEL approach [15] and its variants (RAkEL++ and TREMLC), *i.e.*, the number of models was set to $T = min(2 \times q, 100)$ and a size of labelsets k of 3. These values were found to yield the most satisfactory performances in [15, 79]. The remaining parameters of TREMLC are tuned as suggested by the authors in [79]. In our CkMLC approach, the number of label per bag k was set to 3 as for RAkEL and the committee size m was computed using the following formula: $T = 10 \times ceil(\log(\alpha)/\log(1 - 1/k))$. This formula ensures that each label is drawn 10 times at a confidence level of $\alpha = 1\%$. The *classregtree* Matlab implementation of decision tree was used as the base learner in all compared algorithms. For EBR, ECC and ELP equivalent settings were adopted. The ensemble model were implemented with the bagging strategy [10] to generate diversity within a committee of 100 base-classifiers and the with the *classregtree* Matlab implementation of decision tree as base learner. Finally, instead of manually setting up the single threshold for all labels to 0.5 to output the final bi-partition as in RAkEL and TREMLC, this threshold was tailored to each data set in RAkEL++ using a 5-fold CV procedure [80]. On the other hand, FBR and CkMLC select a separate threshold for each label, using 5-fold CV procedure for FBR and using Oob calibration for CkMLC. We tested 9 different threshold values ranging from 0.1 to 0.9 in 0.1 steps.

We estimate the predictive performance of each compared model using 2-fold cross-validation [88]. To get reliable statistics over the performance metrics, experiments were repeated 25 times. So, the results obtained were averaged over 50 runs. Finally, we wrap up the experiments using statistical tests to evaluate the significant differences among the methods.

4.3.4 Results and Discussion

Detailed average performances of each compared model over the 20 data sets are reported in Tables 4.2-4.9. Each table depicts the results for each analyzed multi-label loss metric. The performances are tabulated in terms of averaged values as well as standard deviations on each data set. The lower the value of the considered metric, the better the algorithm performance is. To examine whether the results are statistically significant, paired t-tests were carried out at 5% significance level. The marker '$\bullet/\circ$' suggests that our approach is statistically superior/inferior to others. Otherwise, a tie is counted and no marker is placed. The obtained $(win/tie/loss)$ counts for CkMLC against the compared algorithms are reported in the bottom row of each table. Furthermore, following [89], if two compared algorithms are, as assumed under the null-hypothesis, equivalent, each should win on approximately $n/2$ out of n data sets. The number of wins is distributed according to the binomial distribution and the critical number of wins at $\alpha = 5\%$ is equal to 15 in our case. Since tied matches support the null-hypothesis we should not discount them but split them evenly between the two classifiers when counting the number of wins; if there is an odd number of them, we again ignore one. Finally, each pairwise comparison for which a variant is significantly better, the $(win/tie/loss)$ count is boldfaced.

In the following, we will first evaluate the performances of the analyzed models over score-based metrics then we will compare the model performances over bi-partition-based metrics.

Performances analysis over *score-based metrics*

Table 4.2 and 4.3 respectively report the models performances over the *Ranking loss* and *One-error*. In order to better assess the effectiveness of our smoothing strategy, we also report the results of our algorithm without smoothing. It will be denoted with the superscript '$*$' in the sequel.

As may be observed over the score-based metrics, CkMLC exhibits the best performances compared to all other algorithms. CkMLC outperforms the other methods by generally achieving the smallest values. This firstly validates the motivation behind our CkMLC method that encouraging diversity in the committee construction achieves more robust votes per label and thus more accurate probability estimates for each label. Moreover, the results also confirm the effectiveness of the smoothing strategy in CkMLC to rank the labels properly. Compared to CkMLC* and

TABLE 4.2: Predictive performances in terms of *Ranking loss*. The lower the score, the better the performance is.

	CkMLC	CkMLC*	RAKEL$_{++}$	TREMLC	RAKEL	fbr$_{M-T}$	EBR	ELP	ECC
Arts	.128±.015	.147±.006•	.126±.003	.139±.006	.733±.024•	.387±.089•	.183±.002•	.135±.003	.143±.003•
Birds	.260±.103	.298±.046	.202±.017	.236±.035	.692±.026•	.255±.043	.356±.027•	.314±.024	.313±.024
Business	.043±.012	.062±.006•	.038±.001	.052±.005	.245±.008•	.183±.049•	.076±.002•	.050±.000	.051±.002•
Computers	.078±.002	.117±.004•	.078±.002	.105±.004•	.483±.011•	.281±.018•	.149±.004•	.105±.004•	.110±.002•
Education	.079±.001	.107±.003•	.080±.001•	.097±.004•	.549±.017•	.401±.034•	.140±.004•	.091±.002•	.099±.003•
Emotions	.158±.010	.159±.009	.213±.016•	.234±.017•	.344±.022•	.373±.022•	.161±.013	.156±.008	.152±.009∘
Enron	.084±.002	.119±.004•	.084±.002	.104±.003•	.367±.012•	.251±.020•	.132±.004•	.105±.002•	.105±.004•
Entertainment	.097±.003	.116±.004•	.102±.002•	.112±.004•	.691±.061•	.376±.030•	.142±.005•	.108±.003•	.112±.003•
Flags	.199±.013	.201±.015	.233±.019•	.255±.018•	.252±.020•	.316±.024•	.225±.013•	.200±.009	.203±.022
Health	.046±.002	.065±.004•	.047±.002•	.060±.003•	.316±.047•	.296±.024•	.088±.004•	.052±.003•	.055±.002•
Image	.147±.008	.150±.007•	.217±.012•	.236±.012•	.264±.015•	.394±.014•	.155±.006•	.147±.007	.143±.007∘
Medical	.029±.007	.056±.014•	.046±.010•	.050±.011•	.187±.015•	.115±.014•	.067±.014•	.041±.007•	.040±.008•
Recreation	.133±.003	.158±.004•	.139±.003•	.153±.006•	.766±.012•	.367±.017•	.200±.006•	.144±.005•	.157±.004•
Reference	.070±.001	.106±.003•	.071±.002	.095±.008•	.446±.010•	.295±.020•	.151±.007•	.084±.002•	.100±.003•
Scene	.075±.003	.076±.003•	.139±.014•	.144±.017•	.137±.008•	.303±.018•	.077±.004	.073±.001	.066±.002∘
Science	.108±.003	.148±.005•	.107±.003	.129±.004•	.619±.016•	.439±.023•	.208±.005•	.125±.004•	.143±.004•
Slashdot	.071±.037	.117±.013•	.061±.006	.093±.013	.201±.011•	.138±.035•	.127±.010•	.092±.007	.099±.007•
Social	.057±.001	.087±.003•	.057±.002	.076±.004•	.277±.007•	.219±.026•	.124±.003•	.074±.002•	.080±.003•
Society	.129±.003	.155±.004•	.128±.003∘	.144±.005•	.522±.014•	.441±.020•	.192±.006•	.147±.003•	.154±.004•
Yeast	.169±.003	.170±.004•	.168±.003	.171±.002•	.266±.005•	.407±.014•	.173±.003•	.172±.003•	.167±.003∘
$(win/tie/loss)$		**(17/3/0)**	(9/10/1)	**(16/4/0)**	**(20/0/0)**	**(19/1/0)**	**(18/2/0)**	**(12/8/0)**	**(14/2/4)**

The marker '•/∘' indicates that CkMLC is significantly better/worse, at a level of significance of 5%. The bottom row reports he obtained $(win/tie/loss)$ counts for CkMLC against the compared algorithms. Bold cells highlight that CkMLC is significantly better than compared algorithm according to the sign test at $\alpha = 5\%$.

TREMLC for which the idea is to mainly encourage the diversity in RAkEL using a triple randomization, the combination of our diverse committee construction and probability smoothing strategy in CkMLC shows promise for obtaining a multi-label k-labelsets framework that enjoys significant improvements in terms of *Ranking Loss* and *One error* metrics.

When compared to classical ensemble models, CkMLC remains competitive and achieves the best performances even if these models (EBR, ELP and ECC) have the advantage to cover all the label space ($\mathcal{L}$) using the same number of models per label. As observed in Table 4.2 and Table 4.3, CkMLC outperforms the ELP model by taking advantage from its reduced complexity and bypass the EBR model by considering the links between the labels in its inner base-models. However, its performances are not statistically distinguishable from the performance of ECC when the *One error* metric is concerned. This is mainly due to the chaining strategy conducted in ECC that also trades off between the label correlation and the label space complexity. The ECC model benefits from the advantage of considering high order correlation by covering all the label space in each base-model (*i.e.* CC here) and also of using the same number of base-models in the majority-voting step. Indeed, ECC works especially well in terms of score-based metrics for data sets having a small number of labels and with a strong conditional dependence between

TABLE 4.3: Predictive performances in terms of *One-error*. The lower the score, the better the performance is.

	CkMLC	CkMLC*	RAKEL$_{++}$	TREMLC	RAKEL	fbr$_{M-T}$	EBR	ELP	ECC
Arts	.266±.035	.488±.010•	.487±.008•	.490±.008•	.485±.006•	.560±.028•	.511±.006•	.481±.007•	.474±.004•
Birds	.268±.026	.419±.061•	.475±.042•	.494±.042•	.379±.083•	.501±.047•	.334±.034•	.334±.029•	.322±.031•
Business	.116±.004	.119±.005•	.121±.005•	.122±.005•	0.12±.005•	.218±.061•	.123±.006•	.122±.005•	.118±.005•
Computers	.382±.008	.381±.010	.392±.010•	.395±.009•	.410±.007•	.487±.016•	.393±.008•	.388±.003•	.375±.005◦
Education	.410±.016	.493±.007•	.495±.007•	.498±.007•	.493±.007•	.578±.010•	.511±.007•	.498±.007•	.487±.007•
Emotions	.274±.022	.276±.025	.341±.029•	.341±.038•	.366±.028•	.326±.033•	.268±.028	.255±.023◦	.253±.017◦
Enron	.233±.008	.231±.004	.237±.007	.242±.006•	.252±.011•	.310±.034•	.229±.008	.229±.005	.217±.006◦
Entertainment	.272±.095	.412±.006•	.424±.010•	.426±.009•	.412±.006•	.502±.006•	.434±.008•	.411±.009•	.404±.010•
Flags	.132±.041	.186±.034•	.242±.048•	.249±.049•	.186±.036•	.233±.060•	.219±.036•	.191±.027•	.201±.033•
Health	.274±.005	.276±.006•	.277±.005•	.277±.006	.328±.018•	.335±.012•	.275±.007	.275±.008	.255±.006◦
Image	.273±.018	.272±.017	.358±.018•	.362±.019•	.308±.012•	.409±.016•	.275±.013	.259±.011◦	.257±.015◦
Medical	.151±.020	.155±.021•	.158±.019•	.163±.021•	.203±.017•	.178±.019•	.126±.013◦	.218±.016•	.198±.018•
Recreation	.191±.016	.473±.007•	.479±.008•	.484±.006•	.477±.006•	.571±.011•	.505±.006•	.468±.007•	.472±.007•
Reference	.393±.008	.390±.010◦	.389±.007◦	.390±.006	.421±.025•	.490±.017•	.409±.008•	.403±.015•	.394±.012
Scene	.227±.009	.228±.009	.314±.019•	.314±.017•	.222±.008	.388±.021•	.219±.008◦	.212±.007◦	.198±.006◦
Science	.403±.014	.526±.010•	.534±.008•	.534±.009•	.528±.009•	.636±.010•	.562±.007•	.532±.010•	.525±.010•
Slashdot	.069±.002	.097±.011•	.111±.008•	.113±.012•	.091±.006•	.148±.020•	.091±.005•	.087±.003•	.087±.003•
Social	.300±.005	.301±.005•	.307±.003•	0.31±.004•	.318±.008•	.374±.008•	.308±.007•	.308±.008•	.299±.008
Society	.418±.012	.427±.011•	.421±.012•	.424±.010•	.423±.016•	.519±.006•	.438±.008•	.434±.009•	.423±.010•
Yeast	.228±.007	.239±.007•	.232±.006•	.233±.008	.235±.007•	.191±.035◦	.221±.007◦	.239±.006•	.232±.007
$(win/tie/loss)$		**(14/5/1)**	**(18/1/1)**	**(17/3/0)**	**(19/1/0)**	**(19/0/1)**	**(13/4/3)**	(12/4/4)	(11/3/6)

The marker '•/◦' indicates that CkMLC is significantly better/worse, at a level of significance of 5%. The bottom row reports he obtained $(win/tie/loss)$ counts for CkMLC against the compared algorithms. Bold cells highlight that CkMLC is significantly better than compared algorithm according to the sign test at $\alpha = 5\%$.

labels, including Emotions, Image, Scene and Yeast data sets (The reader can refer to [18, 21] for more details about the label dependence in these data sets).

Performances analysis over *bi-partition-based metrics*

Tables 4.4-4.9 depict the performances of all compared models in terms of bi-partition-based metrics. In the sequel, the thresholding strategies proposed respectively in CkMLC, FBR and RAkEL++ are implemented separately for each metric. Besides, for the traditional ensemble model EBR, ELP and ECC the majority 0.5 decision threshold is used.

To better assess the effectiveness of our thresholding strategy, we also report, in each table, the results of our algorithm using the majority 0.5 single threshold for all labels. This approach without threshold selection is denoted with the superscript '0.5'.

The results show that CkMLC outperforms both RAkEL and CkMLC$^{0.5}$ that use the single majority threshold 0.5. This validates the motivation behind our threshold calibration strategy to greatly help ensemble multi-label k-labelsets models to reduce bi-partition-based loss metrics.

TABLE 4.4: Predictive performances in terms of *Subset 0/1 loss*. The lower the score, the better the performance is.

	CkMLC	CkMLC$^{0.5}$	RAKEL$_{++}$	TREMLC	RAKEL	fbr$_{M-T}$	EBR	ELP	ECC
Arts	.747±.067	.819±.045•	.891±.035•	.808±.052•	.805±.044•	.880±.022•	.801±.005•	.830±.008•	.819±.009•
Birds	.529±.039	.537±.043	.567±.034	.524±.034	.533±.046	.551±.036•	.501±.011◦	.520±.021	.499±.019◦
Business	.473±.056	.478±.104	.612±.056•	.478±.060•	.490±.095	.583±.021•	.469±.010	.442±.009	.444±.009
Computers	.622±.009	.671±.005•	.814±.009•	.672±.008•	.672±.008•	.756±.006•	.672±.008•	.674±.006•	.664±.005•
Education	.726±.005	.820±.004•	.928±.008•	.800±.004•	.800±.004•	.858±.007•	.794±.004•	.853±.003•	.835±.005•
Emotions	.772±.026	.721±.025◦	.893±.019•	.786±.021•	.786±.021•	.832±.023•	.720±.025◦	.699±.028◦	.688±.017◦
Enron	.869±.010	.887±.005•	.917±.009•	.876±.007•	.876±.007•	.914±.011•	.886±.012•	.885±.008•	.880±.006•
Entertainment	.621±.010	.677±.006•	.820±.015•	.668±.007•	.668±.007•	.786±.003•	.689±.008•	.711±.004•	.699±.007•
Flags	.797±.034	.795±.032	.948±.025•	.803±.041	.803±.041	.894±.018•	.840±.026•	.790±.019	.811±.026
Health	.547±.005	.563±.005•	.793±.011•	.571±.006•	.571±.006•	.738±.012•	.600±.007•	.596±.006•	.574±.005•
Image	.629±.023	.610±.011◦	.858±.008•	.650±.022•	.650±.022•	.740±.014•	.591±.008◦	.606±.010◦	.585±.011◦
Medical	.315±.022	.322±.019	.434±.021•	.314±.015	.314±.015	.369±.021•	.338±.026•	.552±.022•	.640±.014•
Recreation	.689±.004	.754±.006•	.849±.010•	.743±.006•	.743±.006•	.832±.008•	.755±.004•	.795±.006•	.787±.004•
Reference	.561±.005	.635±.006•	.738±.013•	.630±.005•	.630±.005•	.702±.010•	.637±.007•	.661±.007•	.657±.005•
Scene	.494±.018	.481±.012◦	.777±.013•	.513±.018•	.513±.018•	.628±.015•	.461±.012◦	.480±.010◦	.456±.011◦
Science	.740±.005	.839±.007•	.893±.017•	.817±.005•	.817±.005•	.867±.005•	.822±.004•	.883±.005•	.871±.004•
Slashdot	.313±.030	.311±.074	.383±.026•	.323±.031•	.338±.074	.332±.029•	.294±.010◦	.293±.010◦	.294±.008◦
Social	.491±.009	.501±.007•	.665±.016•	.517±.009•	.517±.009•	.631±.010•	.535±.011•	.512±.005•	.511±.005•
Society	.695±.009	.745±.007•	.864±.013•	.753±.008•	.753±.008•	.840±.007•	.741±.006•	.748±.004•	.739±.004•
Yeast	.811±.006	.854±.010•	.955±.009•	.843±.011•	.843±.011•	.956±.004•	.846±.009•	.856±.010•	.841±.006•
$(win/tie/loss)$		(12/5/3)	**(19/1/0)**	**(17/3/0)**	**(15/5/0)**	**(20/0/0)**	(14/1/5)	(13/3/4)	(13/2/5)

The marker '•/◦' indicates that CkMLC is significantly better/worse, at a level of significance of 5%. The bottom row reports he obtained $(win/tie/loss)$ counts for CkMLC against the compared algorithms. Bold cells highlight that CkMLC is significantly better than compared algorithm according to the sign test at $\alpha = 5\%$.

Moreover, the results indicate that diversity in ensemble k-labelsets models is not easy to handle. Indeed, TREMLC achieves disappointing performances since the diversity introduced in the ensemble construction is improved at the expense of the prediction performances of individual multi-label classifiers. On the other hand, the parameter instantiations of TREMLC (the percentage of instance selection and the random subspace size) seem to be more data dependent which tends to deteriorate the performances of the final model [79]. In CkMLC, the diversity effect is managed as long as the model allows repeating several times same k-labelsets and do not use randomization in the feature space.

When compared to classical ensemble models (EBR, ELP and ECC), CkMLC seems to be very competitive and is able to achieve statistically distinguishable performances over multi-label metrics based on F-measure (*i.e. Micro-F1 loss*, *Macro-F1 loss* and *Instance-F1 loss*). However, CkMLC performances are equivalent to these ensemble models over *Subset 0/1 loss*, *Jaccard loss* and *Hamming loss*.

To summarize the obtained results so far, we can draw several conclusions from these observations:

TABLE 4.5: Predictive performances in terms of *Jaccard loss*. The lower the score, the better the performance is.

	CkMLC	CkMLC$^{0.5}$	RAKEL$_{++}$	TREMLC	RAKEL	fbr$_{M-T}$	EBR	ELP	ECC
Arts	.689±.018	.761±.011•	.744±.007•	.708±.031	.866±.013•	.744±.008•	.730±.008•	.797±.009•	.777±.009•
Birds	.448±.016	.471±.017•	.482±.021•	.459±.012•	.481±.019•	.462±.034	.428±.015◦	.503±.022•	.452±.018
Business	.312±.052	.312±.052•	.311±.024	.456±.025•	.297±.003	.375±.005•	.309±.006	.297±.006	.296±.006
Computers	.593±.006	.620±.006•	.619±.008•	.626±.005•	.633±.028•	.628±.007•	.598±.008	.617±.006•	.606±.005•
Education	.677±.006	.789±.006•	.762±.003•	.714±.004•	.819±.011•	.738±.007•	.734±.003•	.828±.004•	.802±.004•
Emotions	.506±.019	.506±.019•	.561±.020•	.660±.037•	.600±.043•	.558±.017•	.482±.018◦	.488±.021◦	.472±.014◦
Enron	.582±.005	.608±.003•	.602±.007•	.597±.007•	.627±.019•	.613±.007•	.557±.016◦	.613±.003•	.569±.006◦
Entertainment	.601±.008	.643±.007•	.628±.007•	.639±.006•	.843±.024•	.666±.005•	.625±.008•	.689±.005•	.666±.008•
Flags	.434±.011	.434±.011•	.425±.029	.476±.028•	.489±.035•	.449±.027	.415±.020◦	.391±.016◦	.394±.017◦
Health	.472±.006	.472±.006•	.474±.005	.572±.007•	.606±.056•	.558±.006•	.468±.004◦	.514±.005•	.477±.004•
Image	.499±.032	.542±.010•	.553±.021•	.736±.037•	.609±.009•	.586±.014•	.499±.007	.540±.012•	.505±.012
Medical	.246±.019	.246±.019•	.241±.016	.279±.016•	.322±.023•	.268±.023•	.249±.023	.494±.024•	.580±.019•
Recreation	.645±.009	.729±.005•	.711±.005•	.683±.005•	.852±.012•	.726±.007•	.705±.006•	.776±.006•	.765±.004•
Reference	.542±.005	.610±.005•	.598±.005•	.580±.004•	.711±.047•	.610±.009•	.589±.007•	.640±.007•	.634±.005•
Scene	.317±.025	.457±.013•	.469±.022•	.786±.053•	.542±.007•	.516±.014•	.426±.012•	.459±.010•	.431±.010•
Science	.688±.009	.817±.007•	.790±.005•	.718±.006•	.866±.019•	.769±.007•	.770±.004•	.872±.006•	.851±.005•
Slashdot	.284±.020	.284±.020•	.256±.017◦	.313±.009•	.291±.012	.258±.017◦	.231±.007◦	.232±.006◦	.233±.004◦
Social	.506±.010	.506±.010•	.468±.007◦	.508±.008	.593±.040•	.520±.008•	.465±.010◦	.482±.004◦	.476±.007◦
Society	.650±.008	.690±.009•	.696±.009•	.706±.008•	.757±.039•	.703±.005•	.656±.007•	.694±.004•	.679±.005•
Yeast	.491±.005	.519±.006•	.513±.008•	.573±.004•	.594±.005•	.587±.007•	.496±.005•	.525±.006•	.503±.005•
$(win/tie/loss)$		**(20/0/0)**	**(14/4/2)**	**(18/2/0)**	**(18/2/0)**	**(17/2/1)**	(9/4/7)	**(15/1/4)**	(12/3/5)

The marker '•/◦' indicates that CkMLC is significantly better/worse, at a level of significance of 5%. The bottom row reports he obtained $(win/tie/loss)$ counts for CkMLC against the compared algorithms. Bold cells highlight that CkMLC is significantly better than compared algorithm according to the sign test at $\alpha = 5\%$.

- CkMLC exhibits the best performances over all the metrics than the original RAkEL and TREMLC.
- The performances of CkMLC are statistically distinguishable from the performance of CkMLC* over score-based metrics. This indicates the effectiveness of our probability smoothing strategy to flatten the label probability distribution and to improve the multi-label classification performances in terms of score-based metrics.
- CkMLC significantly outperforms CkMLC$^{0.5}$ (without threshold calibration) by a noticeable margin over all the metrics (except for *Subset 0/1 loss* and *Hamming Loss*). This confirms the ability of the proposed greedy thresholding algorithm to optimize any performance measure of interest.
- The strategy proposed in CkMLC to calibrate a separate threshold per label seems to perform better than selecting one single threshold for all labels in RAkEL++.
- FBR is worse than CkMLC in all comparisons. Even if the proposed thresholding algorithm has no guarantee of optimality (as for FBR), the results in Tables 4.4 to Tables 4.9

TABLE 4.6: Predictive performances in terms of *Instance-F1 loss*. The lower the score, the better the performance is.

	CkMLC	CkMLC$^{0.5}$	RAKEL$_{++}$	TREMLC	RAKEL	fbr$_{M-T}$	EBR	ELP	ECC
Arts	.580±.004	.737±.038•	.711±.034•	.717±.026•	.855±.014•	.725±.037•	.703±.009•	.785±.009•	.762±.010•
Birds	.414±.023	.443±.015•	.433±.018•	.419±.018	.467±.019•	.465±.026•	.401±.018◦	.496±.023•	.433±.017•
Business	.257±.033	.253±.033◦	.322±.040•	.252±.013	.241±.003	.350±.007•	.252±.006	.244±.006	.243±.005
Computers	.499±.004	.600±.007•	.591±.009•	.599±.008•	.607±.030•	.639±.016•	.570±.009•	.596±.007•	.584±.005•
Education	.570±.006	.778±.006•	.747±.004•	.748±.003•	.809±.012•	.732±.006•	.713±.004•	.820±.004•	.790±.004•
Emotions	.390±.009	.436±.019•	.459±.022•	.486±.020•	.518±.051•	.511±.021•	.404±.017•	.419±.022•	.401±.016
Enron	.451±.006	.501±.004•	.470±.005•	.497±.007•	.528±.020•	.561±.014•	.444±.017	.509±.005•	.459±.007
Entertainment	.503±.008	.630±.007•	.611±.008•	.614±.007•	.837±.024•	.656±.007•	.602±.008•	.681±.005•	.655±.009•
Flags	.281±.009	.292±.019	.305±.017•	.316±.027•	.353±.036•	.544±.075•	.300±.021•	.280±.015	.281±.014
Health	.423±.003	.439±.006•	.429±.004•	.439±.005•	.572±.057•	.523±.011•	.421±.004	.484±.005•	.442±.004•
Image	.433±.008	.519±.010•	.492±.013•	.520±.023•	.469±.035•	.558±.020•	.468±.008•	.517±.013•	.477±.013•
Medical	.209±.013	.220±.019	.215±.016	.216±.016•	.295±.024•	.265±.023•	.219±.023	.474±.025•	.559±.022•
Recreation	.554±.008	.719±.005•	.703±.005•	.699±.005•	.846±.012•	.715±.008•	.686±.007•	.769±.006•	.757±.004•
Reference	.462±.004	.601±.005•	.585±.003•	.587±.004•	.702±.048•	.621±.008•	.572±.007•	.633±.007•	.627±.005•
Scene	.372±.008	.449±.013•	.445±.014•	.454±.023•	.300±.028◦	.493±.011•	.415±.012•	.452±.010•	.423±.010•
Science	.596±.008	.809±.008•	.782±.005•	.780±.006•	.859±.020•	.768±.007•	.751±.005•	.868±.007•	.844±.005•
Slashdot	.217±.038	.258±.027•	.226±.010	.233±.013	.269±.010•	.279±.012•	.210±.006	.211±.004	.212±.003
Social	.390±.005	.449±.008•	.448±.007•	.450±.007•	.583±.042•	.510±.013•	.440±.011•	.471±.004•	.463±.008•
Society	.554±.005	.668±.010•	.651±.010•	.675±.010•	.739±.043•	.688±.007•	.623±.008•	.673±.005•	.655±.005•
Yeast	.414±.005	.411±.005	.395±.004◦	.406±.007◦	.385±.004◦	.502±.010•	.388±.005◦	.417±.005	.396±.004◦
$(win/tie/loss)$		**(16/3/1)**	**(17/2/1)**	**(16/3/1)**	**(17/1/2)**	**(20/0/0)**	**(13/5/2)**	**(16/4/0)**	**(14/5/1)**

The marker '•/◦' indicates that CkMLC is significantly better/worse, at a level of significance of 5%. The bottom row reports he obtained $(win/tie/loss)$ counts for CkMLC against the compared algorithms. Bold cells highlight that CkMLC is significantly better than compared algorithm according to the sign test at $\alpha = 5\%$.

confirm its ability, compared to FBR, to select the relevant thresholds accurately by optimizing the performance measure of interest.

- CkMLC outperforms the other traditional ensemble multi-label methods by generally achieving the lowest values over the used multi-label loss metrics.

4.4 Chapter summary

In this Chapter, we discussed a novel strategy to build and aggregate k-labelsets ensemble multi-label model. The proposed strategy extends and improves upon the original RAkEL algorithm in three ways: i) new randomization strategy using bagging in tandem with random k-labelsets; ii) accounting for the imbalanced label representation when aggregating the base-classifiers predictions; and iii), a specific label threshold calibration procedure on out-of-bag instances.

The proposed ensemble CkMLC approach joins ideas to simultaneously encourage diversity and better aggregate the base-classifiers predictions in tandem with an inner out-of-bag threshold calibration strategy for optimizing a performance measure of interest. Experimental results on

TABLE 4.7: Predictive performances in terms of *Micro-F1 loss*. The lower the score, the better the performance is.

	CkMLC	CkMLC$^{0.5}$	RAKEL$_{++}$	TREMLC	RAKEL	fbr$_{M-T}$	EBR	ELP	ECC
Arts	.582±.022	.681±.011•	.630±.013•	.665±.005•	.662±.009•	.672±.009•	.649±.006•	.726±.009•	.698±.007•
Birds	.587±.034	.699±.054•	.595±.022	.641±.052•	.655±.038•	.605±.028•	.611±.027•	.869±.027•	.706±.013•
Business	.293±.030	.296±.028•	.370±.017•	.290±.009	.296±.026•	.346±.013•	.293±.007	.296±.007	.291±.006
Computers	.484±.007	.551±.006•	.549±.003•	.551±.007•	.551±.007•	.566±.006•	.530±.009•	.552±.008•	.533±.007•
Education	.541±.004	.681±.009•	.612±.003•	.656±.006•	.656±.006•	.653±.006•	.632±.006•	.728±.004•	.701±.005•
Emotions	.366±.017	.363±.016	.400±.008•	.425±.016•	.425±.016•	.421±.017•	.342±.017◦	.346±.014◦	.333±.013◦
Enron	.396±.005	.491±.003•	.474±.006•	.487±.005•	.487±.005•	.488±.005•	.423±.009•	.493±.005•	.442±.005•
Entertainment	.487±.008	.568±.005•	.568±.006•	.551±.007•	.551±.007•	.591±.007•	.542±.008•	.611±.005•	.571±.008•
Flags	.249±.016	.260±.017•	.267±.010•	.276±.025•	.276±.025•	.297±.022•	.266±.016•	.253±.012	.254±.013
Health	.369±.007	.408±.005•	.461±.006•	.407±.006•	.407±.006•	.472±.006•	.402±.004•	.442±.004•	.409±.003•
Image	.413±.013	.428±.009•	.470±.006•	.462±.019•	.462±.019•	.486±.013•	.395±.008◦	.426±.011•	.396±.010◦
Medical	.189±.015	.196±.014	.234±.007•	.194±.012•	.194±.012•	.206±.015•	.191±.016	.369±.020•	.429±.020•
Recreation	.557±.006	.656±.005•	.620±.005•	.638±.007•	.638±.007•	.663±.008•	.634±.007•	.704±.008•	.689±.005•
Reference	.442±.005	.507±.004•	.522±.006•	.500±.004•	.500±.004•	.543±.008•	.502±.006•	.535±.006•	.523±.005•
Scene	.347±.015	.333±.010◦	.418±.007•	.376±.020•	.376±.020•	.418±.012•	.311±.010◦	.332±.007◦	.308±.007◦
Science	.582±.006	.742±.010•	.631±.005•	.713±.008•	.713±.008•	.703±.008•	.690±.003•	.817±.010•	.784±.007•
Slashdot	.229±.014	.225±.041	.325±.014•	.234±.014•	.244±.041	.235±.012•	.215±.007◦	.215±.007◦	.216±.007◦
Social	.387±.004	.406±.010•	.463±.009•	.411±.007•	.411±.007•	.472±.007•	.415±.009•	.423±.006•	.414±.009•
Society	.552±.008	.642±.007•	.621±.008•	.646±.008•	.646±.008•	.654±.005•	.617±.007•	.647±.003•	.630±.004•
Yeast	.330±.004	.380±.005•	.438±.003•	.376±.006•	.376±.006•	.440±.007•	.359±.004•	.386±.005•	.366±.005•
$(win/tie/loss)$		**(16/3/1)**	**(19/1/0)**	**(19/1/0)**	**(19/1/0)**	**(20/0/0)**	**(14/2/4)**	**(15/2/3)**	**(14/2/4)**

The marker '•/◦' indicates that CkMLC is significantly better/worse, at a level of significance of 5%. The bottom row reports he obtained $(win/tie/loss)$ counts for CkMLC against the compared algorithms. Bold cells highlight that CkMLC is significantly better than compared algorithm according to the sign test at $\alpha = 5\%$.

20 benchmark data sets indicate that the proposed model outperforms the RAkEL algorithm and other recent state-of-the-art MLC algorithms over different multi-label loss metrics.

In the next Chapter, further discussions will be conducted to analyze the importance of the combination step as well as effectiveness of our proposed thresholding strategy on different ensemble multi-label classification approaches in order to adapt, in a more principled way, the aggregation procedure to a multi-label performance measure of interest.

TABLE 4.8: Predictive performances in terms of *Macro-F1 loss*. The lower the score, the better the performance is.

	CkMLC	CkMLC$^{0.5}$	RAKEL$_{++}$	TREMLC	RAKEL	fbr$_{M-T}$	EBR	ELP	ECC
Arts	.347±.088	.549±.070•	.567±.024•	.453±.054•	.379±.043	.794±.055•	.531±.036•	.456±.056•	.483±.038•
Birds	.382±.135	.458±.099	.493±.058•	.407±.062	.395±.034	.691±.084•	.374±.058	.436±.071	.465±.086•
Business	.252±.104	.498±.060•	.508±.035•	.308±.092•	.379±.026•	.796±.073•	.435±.047•	.286±.052	.438±.067•
Computers	.244±.044	.518±.023•	.519±.023•	.378±.046•	.548±.026•	.814±.008•	.531±.039•	.336±.045•	.487±.035•
Education	.202±.010	.329±.011•	.328±.012•	.214±.016•	.178±.027◦	.848±.006•	.358±.020•	.235±.021•	.324±.040•
Emotions	.389±.015	.411±.008•	.397±.009	.437±.015•	.520±.008•	.431±.015•	.368±.015◦	.323±.014◦	.320±.013◦
Enron	.175±.020	.394±.031•	.393±.031•	.217±.023•	.444±.027•	.830±.006•	.360±.031•	.398±.034•	.381±.029•
Entertainment	.409±.024	.409±.004	.403±.003	.424±.027	.452±.030•	.776±.008•	.434±.018•	.403±.025	.398±.007
Flags	.304±.054	.322±.011	.315±.014	.337±.049•	.206±.011◦	.371±.026•	.354±.022•	.360±.030•	.304±.014
Health	.228±.034	.310±.017•	.306±.016•	.275±.012•	.207±.024◦	.759±.010•	.333±.031•	.241±.012	.245±.024
Image	.429±.010	.491±.007•	.465±.005•	.461±.021•	.599±.012•	.484±.014•	.400±.009◦	.339±.016◦	.339±.009◦
Medical	.093±.021	.181±.016•	.178±.016•	.112±.022•	.024±.016◦	.628±.023•	.110±.021	.104±.016	.176±.050•
Recreation	.411±.021	.561±.038•	.551±.038•	.507±.037•	.610±.027•	.767±.008•	.567±.051•	.505±.023•	.520±.032•
Reference	.244±.014	.351±.021•	.348±.020•	.285±.029•	.382±.029•	.865±.003•	.341±.031•	.271±.022•	.281±.013•
Scene	.335±.009	.422±.006•	.390±.008•	.368±.019•	.674±.011•	.407±.013•	.316±.009◦	.230±.007◦	.242±.005◦
Science	.310±.029	.480±.021•	.479±.021•	.364±.029•	.532±.027•	.853±.007•	.473±.022•	.418±.030•	.409±.052•
Slashdot	.156±.086	.351±.062•	.360±.044•	.205±.051•	.415±.038•	.825±.151•	.159±.027	.060±.017◦	.210±.037•
Social	.176±.019	.405±.024•	.401±.024•	.249±.040•	.499±.029•	.815±.010•	.364±.053•	.212±.027•	.385±.059•
Society	.323±.022	.532±.041•	.532±.041•	.344±.019	.552±.032•	.853±.006•	.502±.048•	.447±.034•	.520±.047•
Yeast	.391±.022	.483±.003•	.481±.002•	.437±.060•	.486±.003•	.600±.008•	.511±.018•	.445±.004•	.462±.037•
$(win/tie/loss)$		**(17/3/0)**	**(17/3/0)**	**(17/3/0)**	**(14/2/4)**	**(20/0/0)**	**(14/3/3)**	(11/5/4)	**(14/3/3)**

The marker '•/◦' indicates that CkMLC is significantly better/worse, at a level of significance of 5%. The bottom row reports he obtained $(win/tie/loss)$ counts for CkMLC against the compared algorithms. Bold cells highlight that CkMLC is significantly better than compared algorithm according to the sign test at $\alpha = 5\%$.

TABLE 4.9: Predictive performances in terms of *Hamming loss*. The lower the score, the better the performance is.

	CkMLC	CkMLC$^{0.5}$	RAKEL$_{++}$	TREMLC	RAKEL	fbr$_{M-T}$	EBR	ELP	ECC
Arts	.056±.005	.064±.030	.080±.007•	.067±.038	.060±.018	.072±.000•	.055±.000	.055±.000	.054±.000
Birds	.053±.008	.065±.025•	.069±.011•	.054±.010•	.063±.022•	.056±.006•	.048±.002	.050±.002	.046±.002◦
Business	.027±.003	.028±.008	.036±.003•	.027±.003•	.028±.005	.034±.001•	.026±.000	.026±.000	.026±.000
Computers	.036±.000	.035±.000◦	.051±.002•	.036±.000•	.036±.000•	.046±.000•	.035±.000	.035±.000◦	.034±.000◦
Education	.038±.000	.038±.000	.060±.002•	.038±.000	.038±.000•	.050±.000•	.038±.000•	.038±.000•	.038±.000
Emotions	.234±.010	.199±.008◦	.338±.014•	.238±.009•	.238±.009•	.264±.010•	.197±.008◦	.189±.006◦	.187±.007◦
Enron	.046±.000	.047±.000•	.065±.002•	.048±.000•	.047±.000•	.060±.001•	.046±.000	.048±.000•	.046±.000
Entertainment	.053±.000	.052±.000◦	.088±.004•	.053±.001•	.053±.001•	.070±.000•	.054±.000•	.053±.000	.051±.000◦
Flags	.270±.016	.251±.015◦	.314±.013•	.266±.020	.271±.017	.295±.012•	.261±.010	.247±.010◦	.248±.010◦
Health	.034±.000	.033±.000	.053±.001•	.034±.000	.034±.000	.045±.000•	.035±.000•	.035±.000•	.033±.000◦
Image	.186±.005	.160±.003◦	.333±.008•	.197±.010•	.196±.007•	.234±.007•	.164±.004◦	.158±.003◦	.154±.003◦
Medical	.010±.000	.010±.000	.013±.000•	.010±.000	.010±.000	.011±.000•	.011±.000•	.015±.000•	.017±.000•
Recreation	.054±.000	.053±.000◦	.080±.003•	.054±.000•	.054±.000•	.072±.001•	.055±.000•	.055±.000•	.054±.000
Reference	.026±.000	.026±.000◦	.038±.001•	.026±.000	.026±.000•	.035±.000•	.026±.000•	.026±.000•	.025±.000◦
Scene	.114±.005	.095±.002◦	.199±.013•	.117±.007•	.117±.007•	.146±.005•	.091±.002◦	.093±.001◦	.089±.001◦
Science	.032±.000	.032±.000	.047±.001•	.032±.000•	.032±.000	.043±.000•	.033±.000•	.033±.000•	.032±.000•
Slashdot	.016±.001	.017±.005	.021±.001•	.017±.002•	.017±.002•	.018±.001•	.015±.000	.015±.000◦	.015±.000◦
Social	.021±.000	.020±.000◦	.031±.001•	.021±.000•	.021±.000•	.028±.000•	.021±.000◦	.020±.000◦	.020±.000◦
Society	.052±.000	.052±.000◦	.075±.002•	.053±.000	.053±.000	.072±.000•	.052±.000	.053±.000	.052±.000◦
Yeast	.195±.002	.197±.002•	.332±.010•	.197±.003•	.197±.003•	.261±.003•	.195±.002	.198±.002•	.193±.002◦
$(win/tie/loss)$		(3/7/10)	**(20/0/0)**	**(13/7/0)**	**(13/7/0)**	**(20/0/0)**	(7/9/4)	(8/5/7)	(2/5/13)

The marker '•/◦' indicates that CkMLC is significantly better/worse, at a level of significance of 5%. The bottom row reports he obtained $(win/tie/loss)$ counts for CkMLC against the compared algorithms. Bold cells highlight that CkMLC is significantly better than compared algorithm according to the sign test at $\alpha = 5\%$.

Chapter 5

Towards effective aggregation in ensemble multi-label learning

In the previous Chapters, we gave an overview of the different steps in the ensemble multi-label methods and discussed the importance of the combination step in ensemble k-labelsets models. We analyzed how an adequate combination can boost the overall performances of the k-labelsets model. This substantial performance improvement *w.r.t.* ensemble k-labelsets multi-label models, is due to the reflection conducted to make the committee generation consistent with the committee output combination. In this Chapter, we investigate the effectiveness of the combination step and how it influences the prediction performances in traditional ensemble multi-label models.We analyze it from the loss function perspective and distinguish two types of combination schemes, namely ***Label-wise Combination*** and ***Powerset-wise Combination***.

Indeed, ensemble multi-label models consist of a set of multi-label classifiers and present a significant improvement over single multi-label classifier models. This improvement is usually claimed to be attributed to the committee construction and the combination step, with a lack of an in-depth investigation on the conditions under which these steps bring added value.

Even though researchers have designed several ensemble multi-label learning methods [15, 17, 32], they mostly focus on developing strategies for the base-classifier construction and their ability to handle label correlations. Works often propose a new ensemble model and lack a precise study of the combination step and its consistency to the committee construction. Moreover, claimed results in these recently proposed research papers are usually confusing. The authors usually claim that their proposed ensemble model generally outperforms other state-of-the-art approaches in terms of numerous multi-label loss metrics, without specifying the loss metric that the proposed ensemble approach is supposed to optimize.

In general, proposed ensemble multi-label models do often fall short of deepening the understanding of the benefit of ensemble paradigm in the multi-label classification. We rely on several arguments for this, notably the following:

- The combination step is generally carried out in an intuitive manner without formally specifying the output of the ensemble model given the base-classifiers outputs.
- The combination step is seen as a step that improve the predictions without questioning its potential benefits and drawbacks.
- Notions of *label dependence* and *optimized loss function* are considered separately in the base-classifier construction and ignored in the combination step. However, both notions should be considered jointly throughout the various stages of the ensemble multi-label model.

In this Chapter, we aim to elaborate on the base-classifier combination problem. We propose a new formulation for the combination step in the ensemble multi-label models along with a theoretical analysis of the optimized loss function. Our study provides a new perspective on the mechanisms behind the ensemble multi-label models with a deeper understanding of the base-classifiers combination.

In Section 5.1 we formulate two strategies for combining the base-classifiers predictions in ensemble multi-label models: ***(i) Label-wise Combination*** and ***(ii) Powerset-wise Combination***. The latter combination strategy preserves the predicted label structure, whereas the former one considers each label separately and ignores the dependency structure of the label. In Section 5.2, we discuss the influence of the combination strategy on the prediction performances of ensemble multi-label models and highlight the links between the combination strategy and the loss metric optimized by the ensemble model. We present our experimental study in Section 5.3, where experimental results compare several combination strategies on a wide range of multi-label data sets arising from different domains. Finally, we conclude in Section 5.4.

5.1 Multi-label committee combination

In this section, we propose two major strategies for the combination step: *i) The* ***Label-wise Combination strategy*** where the mapping function combines the multi-label outputs separately for each label and *ii) the* ***Powerset-wise Combination strategy*** where the mapping function combines the multi-label outputs jointly as an indivisible information. In the sequel, we first describe the two combination strategies, then denote their main differences.

5.1.1 Label-wise Combination

The ***Label-wise Combination*** strategy is the most popular combination scheme in ensemble multi-label models. It considers each label independently, such that the base-classifiers outputs are combined for each label separately. In other words, to decide for the ensemble prediction, the base-classifiers outputs are averaged for every label. Thus, the ensemble output is a vector of probability scores (a probability score for each label $s^i(\mathbf{x})$) indicating the relevance of each label $\lambda_i \in \mathcal{L}$ for the predicted instance. Each label score $s^i(\mathbf{x})$ is the average of the committee predictions for the label λ_i. Depending on the information provided by the base-models, we define the ***Label-wise Combination*** (***LC***) strategy of an ensemble multi-label model $H = \{h_1, \cdots, h_T\}$ as follows:

- If each base-model h_t provides a vector of crisp label predictions $h_t^i(x) \in \{0; 1\}$ with $1 \leq i \leq q$, the ***LC*** strategy is formulated as :

$$\mathbf{Sl}(\mathbf{x}) = (Sl^1(\mathbf{x}), \cdots, Sl^q(\mathbf{x})) \; : \; Sl^i(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} h_t^i(\mathbf{x})$$

- If each base-model h_t provides a vector of label probability score $s_t^i(x) \in [0; 1]$ with $1 \leq i \leq q$, the soft version of the ***LC*** strategy is formulated as :

$$\mathbf{Ss}(\mathbf{x}) = (Ss^1(\mathbf{x}), \cdots, Ss^q(\mathbf{x})) \; : \; Ss^i(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} s_t^i(\mathbf{x})$$

The meaning of the predicted scores is different according to the base-models output. When combining crisp labels prediction ($h_t^i(\mathbf{x}) \in \{0, 1\}$) the scores given by the ***LC*** estimates the probability that a multi-label model (a multi-label classifier in this case) assigns the label λ_i giving the instance $\mathbf{x}$: $Sl^i(\mathbf{x}) \simeq p(h^i(\mathbf{x}) = 1|\mathbf{x})$. On the other hand, when combining probability output ($s_t^i(\mathbf{x}) \in [0, 1]$) the ***LC*** result estimates the probability to assign the label λ_i given the instance $\mathbf{x}$: $Ss^i(\mathbf{x}) \simeq p(y^i|\mathbf{x})$.

5.1.2 Powerset-wise Combination

The ***Powerset-wise Combination*** strategy considers jointly the information predicted by each base-classifier and produces a probability score for each labelset in $\mathcal{Y}$. Depending on the information provided by the base-classifiers, we define the ***Powerset-wise Combination*** (***PC***) strategy of an ensemble of multi-label models $H = \{h_1, \cdots, h_T\}$ as follows:

- If each base-model h_t provides a vector of crisp label predictions (*i.e* $\mathbf{h}_t(\mathbf{x}) \in \mathcal{Y}$) the ***PC*** computes the frequency of each labelset over the ensemble committee predictions as :

$$\forall \mathbf{y} \in \mathcal{Y} \ , \ Sl^{\mathbf{y}}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} I(\mathbf{h}_t(\mathbf{x}) = \mathbf{y})$$

- If each base-model h_t provides a label probability score for each labelset $s_t^{\mathbf{y}}(\mathbf{x}) \in [0; 1]$ with $\mathbf{y} \in \mathcal{Y}$, the soft version of the ***PC*** strategy averages the predicted distribution over the committee as :

$$\forall \mathbf{y} \in \mathcal{Y} \ , \ Ss^{\mathbf{y}}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} s_t^{\mathbf{y}}(\mathbf{x})$$

As in the ***LC*** strategy, the meaning of the estimated scores is different according to information provided by the base-models. When combining crisp label output ($\mathbf{h}_t(\mathbf{x}) \in \{0, 1\}^q$) the score Sl resulting from the ***PC*** strategy estimates the probability that a multi-label classifier assigns the labelset $\mathbf{y}$ given the instance $\mathbf{x}$: $Sl^{\mathbf{y}}(\mathbf{x}) \simeq p(\mathbf{h}(\mathbf{x}) = \mathbf{y}|\mathbf{x})$, with $\mathbf{y} \in \mathcal{Y}$. On the other hand, when each base-model provides $s^{\mathbf{y}}$ as an estimation of $p(\mathbf{y}|\mathbf{x})$ for each labelset $\mathbf{y} \in \mathcal{Y}$; the ***PC*** strategy estimates the probability to assign the labelset $\mathbf{y}$ given the input $\mathbf{x}$: $Ss^{\mathbf{y}}(\mathbf{x}) \simeq p(\mathbf{y}|\mathbf{x})$, with $\mathbf{y} \in \mathcal{Y}$.

5.1.3 *Label-wise Combination* Vs *Powerset-wise Combination*

LC strategy is the commonly used combination in ensemble multi-label models [16, 17, 32] due to its simplicity and low computational cost. As the ***LC*** strategy considers each label separately, the probabilistic dependency structure of the labels (given the inputs) is ignored in the combination step. Furthermore, the ***LC*** strategy may potentially break the labels' structure learned by the individual base-classifiers which leads to failures in predictions. In contrast, by considering the predicted labels as an indivisible entity, the ***PC*** preserves the labels' structure predicted by each base-classifier in the ensemble committee. However, the ***PC*** strategy suffers an important computational complexity, being exponential with the number of labels. Nevertheless, it is worth mentioning that labelsets present in the data set are consistently dominated by a small minority of core label combinations. This prevalent character in multi-label data sets makes the use of ***PC*** easier when coupled with an appropriate multi-label base-learner despite the exponential number of possible labelsets. Furthermore, this complexity can be bypassed by adopting a crisp label formulation of the combination strategy. Notice that the soft combination is generally used only for a homogeneous committee. For the heterogeneous committee, the probabilities generated by the different types of base-classifiers cannot be aggregated without a careful calibration. In such situations, the predicted probabilities are often converted to crisp labels, and then a crisp label combination strategy is applied [86].

5.2 Ensemble multi-label combination and loss metrics

As aforementioned, two strategies are possible for combining base-classifiers predictions in an ensemble model, and each addresses the relationship between labels in a different way. Thereby, two important questions remain: *i*) Does the combination strategy influence the predictions of the ensemble model ? and, *ii*) How to make the combination step consistent with the loss function optimized by the base-classifiers ?

In this Section, we throw light on these questions and give a first theoretical insight on the cost optimized by each combination strategy, and we also discuss the loss consistency in ensemble multi-label models.

5.2.1 Why different combination strategies ?

It is evident from a Bayesian perspective that for a committee of Bayes-optimal predictors, there is no need to distinguish between the ***LC*** and ***PC*** strategies since, regardless of the combination strategy, the final prediction will be the same. If we suppose that the ensemble committee is formed with duplicates of the optimal classifier h^*, all the base-classifiers will predict the same labelset. Under this optimal conditions, one can alternatively think of selecting any predictor output to get a correct optimal prediction, regardless of the objective loss metric.

Moreover, for an objective label-wise decomposable metric, a correct prediction for the committee can also be constructed by selecting for each label λ_i, a random prediction within the predictor outputs for λ_i. In numerical experience, this typically occurs when a labelset dominates the predictions of all the predictors outputs with a probability greater than 0.5. In this case, demonstrating the equivalence is simple. Dembczyński *et al.* [18] proved a very similar result, although the proof turns out to be much simpler in this context.

Proposition 2. The ***LC*** and ***PC*** strategies have the same predictions, i.e., $\mathbf{H}\text{-}PC(\mathbf{x}) = \mathbf{H}\text{-}LC(\mathbf{x})$, if the probability of the mode of the base-classifier labelset output is greater than 0.5, i.e., $p(\mathbf{H}\text{-}PC(\mathbf{x})|\mathbf{x}) > 0.5$.

Proof. Since the probability of the jointly combined labelset $\mathbf{H}\text{-}PC(\mathbf{x}) = \ell$ is greater than 0.5, *i.e.*, $p(\ell|\mathbf{x}) > 0.5$, the marginal probabilities of $\lambda_i \in \ell$ or can be written by: $p(\lambda_i|\mathbf{x}) = p(\ell|\mathbf{x}) + \sum_{\ell' \in \mathcal{P}(\mathcal{L} \setminus \lambda_i)} p(\lambda_i \cup \ell'|\mathbf{x})$ and is always greater than 0.5. The statement also holds for $\lambda_i \in \mathcal{L} \setminus \ell$. Thus, the joint mode is decomposed on marginal modes and we have $\mathbf{H}\text{-}LC(\mathbf{x}) = \mathbf{H}\text{-}PC(x)$. □

This result points out a misleading situation where the usefulness of the distinction between the two combination strategies is challenged.

However, it is important to notice that if we could build such perfect machine learning model, which would give every time the best possible prediction by sheer force, there will be no need of a committee model itself (since it is only a set replicate of the same predictor), neither for ensemble learning paradigm in general. Furthermore, when the numerical equivalence condition holds, the resulting committee is made of many strong predictors where the predictions are highly correlated. In this context, the ensemble approach will not necessarily lead to a significant performance improvement and it would be better to use a single multi-label model [10, 90]. Moreover, the diversity behind the committee construction is acting against creating duplicates base-classifier and aims to create dependent predictors. The diversity in the committee construction is expected to produce a flat distribution over the labelset predictions where the correct labelset is taking the largest score, rather than a sharp distribution on (or near [1]) the correct labelset. And thus, in the multi-class equivalent setting of the multi-label task (the case where ***PC*** strategy operates and also where the equivalence condition is verified), there is no guarantee that the majority class is predicted more than 50% (absolute majority).

The underlying principle of ensemble paradigm is a recognition that in real-world situations, every model has limitations and will make errors. Within these "limitations", the purpose of ensemble learning is to trade-off their strengths and weaknesses, heading to the best possible overall predictions being taken [91]. Thus the combination should be conducted in order to enhance the prediction performance. Several theoretical and empirical works have demonstrated that ensemble model can significantly overtake single model in terms of the overall prediction accuracy [10, 11, 91].

In a frequentist perspective, this is motivated considering the trade-off between *bias* and *variance*, which decomposes the model error into two components.

Namely the *bias* component and *variance* component, where the *bias* component results from the difference between the estimated model and the actual one and the *variance* component expresses the model sensitivity regarding the individual data points. Indeed, when training multiple models, and then averaging the resulting predictions, the contribution arising from the *variance* component tended to cancel, leading to improved predictions. As the ***LC*** strategy considers each label separately, the probabilistic dependency structure of the labels (given the inputs) is ignored in the combination step.

However, this may help the committee to avoid considering pointless variability due to the data noise and thus achieve more accurate prediction for each label. On the other hand, by considering the predicted labels as an indivisible information, the ***PC*** strategy preserves the labels' structure predicted by each base-classifier. Thus, it allows the committee to consider the inherent variations within the labelsets predicted by the base-models and draws near the optimal labelset. The difference between the two combinations strategies is blatant when there are multiple modes or

[1] near in terms of similarity between the associated labels: almost the same subset of associated labels.

multiple optimums. In the following, we throw light on the theoretical evidence supporting our distinction between the two combination strategies.

5.2.2 Toward theoretical insights into multi-label combination

The final decision of a multi-label ensemble classifier $H = \{h_1, \cdots, h_T\}$ when using the ***LC*** strategy is obtained by the popular majority voting of the outputs received by each label separately from each ensemble member. Depending on the nature of the outputs from each ensemble member, this combination is formulated as follows:

- If the committee members output crisp labels $h_t^i(\mathbf{x}) \in \{0; 1\}$ with $1 \leq i \leq q$:

$$\mathbf{H}(\mathbf{x}) = (H^1(\mathbf{x}), \cdots, H^q(\mathbf{x})) \; : \; H^i(\mathbf{x}) = \underset{y^i \in \{0,1\}}{\operatorname{argmax}} \sum_{t=1}^{T} I(h_t^i(\mathbf{x}) = y^i)$$

- If the committee members output label probability scores $s_t^i(\mathbf{x}) \in [0; 1]$ with $1 \leq i \leq q$:

$$\mathbf{H}(\mathbf{x}) = (H^1(\mathbf{x}), \cdots, H^q(\mathbf{x})) \; : \; H^i(\mathbf{x}) = \underset{y^i \in \{0,1\}}{\operatorname{argmax}} \sum_{t=1}^{T} s_t^i(\mathbf{x})$$

As a result, it follows that ***LC*** strategy is well suited for every loss metric whose risk-minimizer can be expressed marginally. Moreover, the risk-minimized by the ***LC*** rule is exactly the *Hamming loss* risk-minimizer, when the ***LC*** decision rule combines the base-classifiers' estimated probability distributions $s^i(\mathbf{x})$. Indeed, the *Hamming loss* risk-minimizer is formulated in [18] as:

$$\mathbf{h}^*(\mathbf{x}) = (h^{*1}(\mathbf{x}), \cdots, h^{*q}(\mathbf{x}))$$

where

$$\mathbf{h}^{*i}(\mathbf{x}) = \underset{\mathbf{y}^i \in \{0,1\}}{\operatorname{argmax}} \, p(y^i | \mathbf{x})$$

More generally, when the ***LC*** decision rule combines base-classifiers outputs and the loss metric optimized by the base-models is label-wise decomposable. Thus, the committee and the base-models optimize the same loss metric (*i.e.* $L_{H\text{-}LC} = L_h$).

Certainly, assuming that each base-model $\mathbf{h}$ outputs the optimal prediction over the metric L_h, if L_h is label-wise decomposable, it follows that $\mathbf{h}^i(\mathbf{x})$ is optimal for L_h over each label λ_i : $1 \leq i \leq q$. Indeed, since that the ***LC*** decision rule selects the most frequent prediction within the committee's predictions separately for each label λ_i, the committee prediction ($Hl^i(\mathbf{x}) =$

$\underset{y^i \in \{0,1\}}{\text{argmax}} \sum_{t=1}^{T} I(h_t^i(\mathbf{x}) = y^i)$), also gives the optimal prediction for L_h across each label λ_i. Thus, via the label decomposition of the L_h, $Hl(\mathbf{x})$ is also optimal for the L_h.

Thereby, the ***LC*** decision rule cannot be adequate for instance-wise loss metrics like the *Subset 0/1 loss*, the *Instance-F1 loss* or the *Jaccard loss*.

To decide about the ensemble output estimated on a ***PC*** strategy, the majority vote considers jointly the predicted labelset. The ensemble output is the labelset predicted by the largest number of base-models or the labelset with largest average score. Hence, the ***PC*** step and the majority vote step of a committee are written as follows:

$$\mathbf{H}(\mathbf{x}) = (mode\{\mathbf{h}_1(\mathbf{x}), \cdots, h_7(\mathbf{x})\})$$

We define the *Powerset-wise Combination decision rule* of an ensemble of classifiers $H = \{h_1, \cdots, h_T\}$ as follows :

- If the base-models provide crisp labels $\mathbf{h}_t(\mathbf{x}) \in \mathcal{Y}$:

$$\mathbf{Hl}(\mathbf{x}) = \underset{\mathbf{y} \in \mathcal{Y}}{\text{argmax}} \sum_{t=1}^{T} I(\mathbf{h}_t(\mathbf{x}) = \mathbf{y})$$

- If the base-models provide labelsets probability scores $s_t^{\mathbf{y}}(\mathbf{x}) \in [0; 1]$ with $\mathbf{y} \in \mathcal{Y}$:

$$\mathbf{Hs}(\mathbf{x}) = \underset{\mathbf{y} \in \mathcal{Y}}{\text{argmax}} \sum_{t=1}^{T} s_t^{\mathbf{y}}(\mathbf{x})$$

It follows that ***PC*** is most suitable for the class of multi-label loss functions that require the joint label prediction $\mathbf{y}$ or the estimation of the joint conditional probability distribution in the case of the soft combination $p(\mathbf{y}|\mathbf{x})$.

Furthermore, the risk-minimized by a committee of base-models estimating the $p(\mathbf{y}|\mathbf{x})$ combined via the ***PC*** decision rule is exactly the *Subset 0/1 loss* risk-minimizer [18] which is :

$$\mathbf{h}^*(\mathbf{x}) = \underset{\mathbf{y} \in \mathcal{Y}}{\text{argmax}} \sum_{t=1}^{T} p(\mathbf{y}|\mathbf{x})$$

When aggregating crisp labels, the majority decision rule coupled with the ***PC*** strategy is an estimation of the optimal prediction regardless of the type of loss metric optimized by the base-models. Indeed, assuming that each base-classifier outputs $\mathbf{h}(\mathbf{x})$ is optimal for L_h, the most frequent labelset, within all the base-models predictions, selected by the ***PC*** decision rule is necessarily optimal for L_h.

5.2.3 Loss function consistency in ensemble multi-label models

In previous sections, a link between the combination strategy and multi-label loss metrics is established and we showed that the combination strategy is firmly connected to the function optimized by the ensemble model. As a meta-algorithm, ensemble models can be built on top of any multi-label learner themselves optimizing a particular loss function [18]. However, one cannot build an ensemble multi-label model using a multi-label base learner optimal for a specific metric and then thoughtlessly combine the committee outputs (or vice versa). For instance, building a committee of base-classifiers that learn each label separately, then combining the base-classifier predictions using the ***PC*** strategy to be optimal for an instance-wise metric. In fact, such ensemble construction will lead to output labelsets that are inadequate or impossible for the task and in any case optimal for a well-defined loss metric such as the *Subset 0/1 loss*. Thus, it is important to build an ensemble model where the combination step is in line with the loss function optimized by the base-learners (and vice versa). Therefore, it is better to set first the objective function to optimize by the ensemble model then, determine the adequate base-learner and the compatible combination strategy jointly. In other words, to be optimal for label-wise decomposable measure (respectively for instance-wise decomposable measure) it is more appropriate to combine base-classifier that optimize label-wise decomposable (respectively instance-wise decomposable) multi-label performance measures using ***LC*** strategy (respectively using ***PC*** strategy). Recall that instance-wise decomposable measure are not label-wise decomposable measure.

However, the ***PC*** strategy has never been recommended in the literature despite proposing ensemble models considering the links between the labels such as in ECC [17], ELP [16] and RFPCT [32].

Besides, in some situation one can be only reluctant for the actual loss optimized by some base-classifiers such as in (RFPCT and VPCME). Furthermore, in some other contexts the objective loss function may not have a known optimal multi-label model (such as in the case of the *Jaccard loss* [18]). In such case, it is essential to give a brand-new meaning to the loss optimized by this ensemble multi-label whatever the base-classifier is. The straightforward option in this case is to use the threshold calibration.

On the other hand, as ensemble models use the bagging strategy to generate their committee, it is promising to take advantage from the ensemble construction step to build a parallel out-of-bag calibration data set. Thus, we propose to use our out-of-bag *Forward Multi-label Thresholds Calibration* algorithm presented in Chapter 4.

The proposed optimization algorithm is valid for all ensemble models based on the bagging strategy. To the best of our knowledge, this is the first attempt to propose an algorithm for selecting a distinct threshold per label by optimizing any multi-label performance measure of interest for ensemble multi-label models. All the more, since the threshold calibration is independent from the

base-classifier generation, it will allow the same committee to be used -alongside with different thresholds- to achieve appropriate prediction across different metrics.

In the next Section, we will analyze the behaviour of ensemble multi-label model with respect to the two combination strategies. We will also highlight the benefit of our proposed out-of-bag threshold calibration and how it can be used to tweak ensemble multi-label predictions across different metrics.

5.3 Experimental evidence

To substantiate the theoretical results by means of empirical evidence, this Section presents an experimental analysis of numerous ensemble models coupled with both ***LC*** and ***PC*** strategies over a wide range of multi-label data sets. We first describe the experimental design, then we describe the data sets used in this study. Next, we state the parameter settings for all compared ensemble multi-label algorithms. Finally, we present and discuss the experimental results.

5.3.1 Experimental design

Our aim in this empirical analysis is not to conduct a comprehensive comparison of the existing multi-label ensemble methods in the literature, but to understand the influence of the combination strategy over the ensemble multi-label performances. We hope to provide useful insights into the link between the combination strategy within the ensemble approach and the optimized loss function.

Thus, we first evaluate the performance of each combination strategy in each ensemble approach over different multi-label loss metrics. In a second time, the best performing combination strategy for each ensemble approach on each metric are selected and compared together.

This study explores these questions for six ensemble multi-label methods including both "*problem transformation*" and "*algorithm adaptation*" ones. The set of compared models consists of *Ensemble of Binary Relevance* model (EBR) [3, 17], *Ensemble of Label Powerset* model (ELP), [16, 27], *Ensemble of Classifier Chains model* (ECC) [17], *Random Forest Predictive Clustering Tree* (RFPCT) [32], *RAndom k-labELsets* (RAkEL) [15], *Variable Pairwise Constraint projection for Multi-label Ensemble* (VPCME) [77] and our proposed *Calibrated k-labelsets Multi-Label Classifier* CkMLC presented in Chapter 4.

The compared approaches come in four variants. The first variant corresponds to the ***Label Combination*** strategy and is suffixed with '$-LC$' while the second variant corresponds to ***Powerset Combination*** strategy and is suffixed with '$-PC$'. As aforementioned, these variants correspond

to the majority voting strategy: $-LC$ for label majority vote and $-PC$ for labelset majority vote. The other two variants are a specific to ***Label Combination*** strategy and correspond to the two threshold calibration variants (Single-threshold and Multi-threshold). Variants with Single-threshold strategy (respectively with Multi-threshold) are denoted with the subscripts '$-LC_{S-T}$' (respectively with '$-LC_{M-T}$'). These two former variants are examined in order to shed some further light on the differences observed when threshold calibration is performed for optimizing a multi-label indicator. In the Multi-threshold strategy, a finely-tuned threshold is associated to each label based on our *Forward Multi-label Thresholds Calibration* algorithm instead of using a single tuned threshold for all the labels in the Single-threshold strategy. On the other hand, it is noteworthy that RAkEL and CkMLC base-classifiers cannot be aggregated using the ***PC*** strategy since their base-classifiers do not predict all labels (see Section 3.3.2).

To assess the effectiveness of the different combination strategies and performances of the analyzed methods, we conducted the experiments on 20 benchmark data sets from the *Mulan's repository* [87]. The selected data sets were broadly used in various studies on multi-label learning and cover different application domains: biology, semantic scene analysis, music emotions and text categorization. Table 5.1 summarizes the main statistics of these data sets: the number of features **M**, the number of labels **q**; the Label Cardinality ***Card***$= \frac{1}{N}\sum_{i=1}^{N}|Y_i|$, which is the average number of labels associated with each example; the Label Density **LD**$= \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i|}{q}$, which is the normalized ***Card***.

On the other hand, algorithm's performances were analyzed according to six commonly used multi-label performance measures including *Subset 0/1 loss*, *Jaccard loss*, *Instance-F1 loss*, *Micro-F1 loss*, *Macro-F1 loss* and *Hamming loss*. The selection of these measures was made toward analyzing the performances of all compared approaches on both label-wise decomposable (*Hamming loss*, *Macro-F1 loss*) and instance-wise decomposable (*Subset 0/1 loss*, *Jaccard loss*, *Instance-F1 loss*) metrics (See Section 2.3).

5.3.2 Experimental setup

To make fair analysis, the same ensemble size $T = 100$ was adopted for all the compared methods, except for RAkEL and CkMLC where the committee size depends on the label space cardinality $|\mathcal{L}|$ [15]. Thus, the ensemble size for RAkEL was set to $T = min(2q, 100)$ [15] k was set to 3. For the CkMLC approach, the number of labels per bag k was set to 3 as for RAkEL and the committee size m was computed using the following formula: $T = 10 \times ceil(\log(\alpha)/\log(1 - 1/k))$. The diversity within the committee of 100 base-classifiers is generated using the bagging strategy [10]. The *classregtree* Matlab implementation of decision tree was used as the base learner for EBR, ECC, ELP, RAkEL and CkMLC. Besides, as suggested by authors in [77], the instance-based learning method MLkNN with $k = 10$ [5] was used for VPCME due to its excellent

TABLE 5.1: Description of the multi-label data sets used in the experiments.

Data	**Domain**	**N**	**M**	**q**	***Card***	**LD**
Arts	Yahoo-Text	5000	462	26	1.636	0.063
Birds	Audio	645	260	19	1.014	0.053
Business	Yahoo-Text	5000	438	30	1.588	0.053
Computers	Yahoo-Text	5000	681	33	1.508	0.046
Education	Yahoo-Text	5000	550	33	1.460	0.044
Emotions	Music	593	72	6	1.869	0.311
Enron	Text	1702	1001	53	3.378	0.064
Entertainment	Yahoo-Text	5000	640	21	1.420	0.068
Flags	Image	194	19	7	3.392	0.485
Health	Yahoo-Text	5000	612	32	1.662	0.052
Image	Image	2000	249	5	1.236	0.247
Medical	Text	978	1449	45	1.245	0.028
Recreation	Yahoo-Text	5000	606	22	1.423	0.065
Reference	Yahoo-Text	5000	793	33	1.169	0.035
Scene	Image	2407	294	6	1.074	0.179
Science	Yahoo-Text	5000	743	40	1.540	0.036
Slashdot	Text	3782	1079	22	1.180	0.041
Social	Yahoo-Text	5000	1047	39	1.283	0.033
Society	Yahoo-Text	5000	636	27	1.692	0.063
Yeast	Biology	2417	103	14	4.237	0.303

predictive performance. Besides, its variable pairwise constraint threshold was set to 0.6 as recommended in [77]. Furthermore, we investigate the behavior of all ensemble models within *crisp label aggregation* since that some models can not predict the probability distribution over all possible labelsets $s^{\mathbf{y}}(\mathbf{x})$.

Notice that the soft combination is generally used only for a homogeneous committee. For the heterogeneous committee, the probabilities generated by the different types of base-classifiers cannot be aggregated without a careful calibration. In such situations, the predicted probabilities are often converted to crisp labels, and then a crisp label combination strategy is applied [86].

For both Single-threshold and Multi-threshold strategies, different threshold values ranging from 0.1 to 0.9 in 0.1 steps were considered in the calibration step as in [17]. For both strategies, out-of-bag instances are used as an unbiased validation set and Algorithm 3 is performed for the Multi-threshold strategy. As aforementioned before, RAkEL and CkMLC base-classifiers cannot be aggregated using the ***PC*** strategy. For these two models, only three variants are reported the '$-LC$' variant, the '$-LC_{S-T}$' variant and the '$_{M-T}$' variant.

Moreover, we estimate predictive performances by using 2-fold cross-validation [88]. To get reliable statistics over the performance metrics, experiments were repeated 25 times. So, the results obtained were averaged over 50 iterations. Finally, we wrap up the experiments using statistical tests to evaluate significant differences among methods.

5.3.3 Results and discussion

Detailed average performances of each ensemble version for all 20 data sets using the protocol described above are reported in Tables A.1-A.6 in the Appendix. Each table depicts the models performances in terms of each considered multi-label loss metric. Models performances are tabulated in terms of averaged values as well as standard deviations for each ensemble variant and over each data set.

To help summarize the results, we conduct statistical analysis to better assess the results obtained for the different variants of each ensemble algorithm on each metric. Thus, we adopt in this study the methodology proposed by [89] for the comparison of several algorithms over multiple data sets. In this methodology, the non-parametric Friedman test is firstly used to evaluate the rejection of the hypothesis that all the classifiers perform equally well for a given risk level (i.e. in our case all the ensemble version are equally well for a given risk level). It ranks the algorithms for each data set separately, the best performing algorithm getting the rank of 1, the second best rank 2 etc. In case of ties it assigns average ranks. Then, the Friedman test compares the average ranks of the algorithms and calculates the Friedman statistic. If a statistically significant difference in the performance is detected, we proceed with a *post-hoc* test. The Nemenyi test is used to compare all the methods to each other. In this procedure, the performance of two methods is significantly different if their average ranks differ more than some critical distance (CD). The critical distance depends on the number of algorithms, the number of data sets and the critical value (for a given significance level p) that is based on the Studentized range statistic (see [89] for further details).

In this study, the Friedman test reveals statistically significant differences ($p < 0.05$) between the ensemble version and over for all the performance measures. One case do the exception (EBR over the *Subset 0/1 loss*) we will highlight it when discussing its specific results. Furthermore, we present the result from the Nemenyi post-hoc test with average rank diagrams as suggested by Demsar [89]. These are given on Figures 5.1 - 5.7. The ranks are depicted on the axis, in such a manner that the best ranking algorithms are at the rightmost side of the diagram. The algorithms that do not differ significantly (at $p = 0.05$) are connected with a line. The critical difference CD is shown above the graph.

As may be observed in Figures 5.1- 5.7 and Tables A.1-A.6, the ***LC*** strategy is significantly better than the ***PC*** one over label-wise metrics (*Hamming loss* and *Macro-F1 loss*). On the other

hand, the ***PC*** strategy is significantly better than the ***LC*** one over instance-wise metrics (*Subset 0/1 loss*, *Jaccard loss* and *Instance-F1 loss*). This advantage is more pronounced when the label correlation is considered in the training process (in the case of ELP and ECC).

As far as the ELP model is concerned, results in Figure 5.2 and Tables A.1-A.6, corroborate our previous finding, namely that the ***PC*** strategy is well-tailored for *Subset 0/1 loss* minimization in ELP. Obviously, the loss function minimized by both ELP's base-classifier (LP here) and the ***PC*** strategy is the *Subset 0/1 loss*. This confirms that preserving the coherence of the optimized loss function throughout the ensemble model construction (base classifier generation + combination) may yield a high improvement in performance. The same observation also holds for ECC-***PC*** results on the *Subset 0/1 loss*, since that the loss function minimized by the ECC's base-learner (CC) is the *Subset 0/1 loss* [18]. More generally, results assert that the ***PC*** strategy preserve the quality of the predicted labelsets by the base-classifiers, and thus the ***PC*** strategy is generally suitable for models that aim to learn a join label distribution directly such as ELP or via heuristics such as ECC or RFPCT.

Besides, the ***PC*** strategy, as expected, is arguably inefficient when coupled with a multi-label base-classifier that ignores the inter-dependencies between the labels in the training process as in the EBR where variants performances are not distinguishable. Indeed, ***PC*** seems to be the worst performing methods for EBR on all metrics, except for *Subset 0/1 loss*, for which no clear conclusion emerged when one examines their values in Table A.1 and A.2.

On the other side, the ***LC*** strategy slightly improves the results of models considering the links between the label (completely such as in ELP or in an approximate way such as in ECC and RFPCT) compared to the ***PC*** strategy on label-wise loss metrics. An effect that could be attributed to the ability of ***LC*** to correct the prediction made for each label based on the agreement of the base-classifier on each individual label. Moreover, ***LC*** strategy achieves the best performing performances with the EBR over the *Hamming loss* since that loss function remains consistent within the ensemble model.

Another interesting observation when looking at the average rank diagrams is that calibrated variants (-LC_{S-T} and -LC_{M-T}) are dominating all the models whatever the analyzed metrics, meaning that the threshold calibration is beneficial for the multi-label models. We found calibration to be remarkably effective at improving the performance of all the models over all the metrics compared to the majority-voting based approaches. The models performances are significantly improved when the loss function optimized by the base-classifiers is different from the metric of interest such as in VPCME.

Moreover, the -LC_{M-T} strategy used to calibrate a separate threshold per label seems to perform better than calibrating one single threshold for all labels (*i.e* -LC_{S-T}). In general, the LC_{M-T}

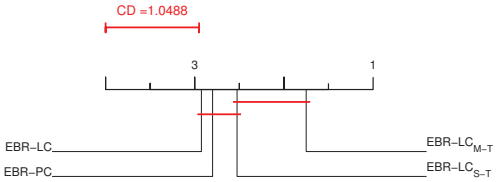
variant exhibits the highest performances in terms of all metrics and seems to be the more suitable strategy for models that ignores the labels correlations such as EBR. We also note that the calibrated variants achieve equivalent performance to those obtained with optimal variants. Indeed, we observe that the performances of EBR-LC_{M-T} and EBR-LC are not distinguishable over the *Hamming loss* as well as the performances of ELP-LC_{M-T} and ELP-PC over the *Subset 0/1 loss* which makes the use of calibrated variants valid for all metrics.

Over k-labelsets ensemble models (*i.e* RAkEL and CkMLC), the best performing variant is the calibrated variant over all metrics. This rolls out that their performances are generally boosted when the thresholds are calibrated and illustrates the effectiveness of the calibration step for the model proposed in Chapter 4.

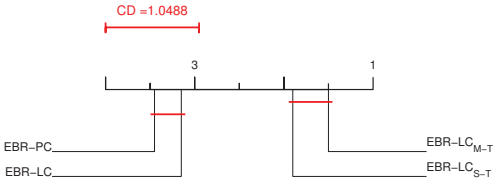
To briefly summarize the obtained results, we draw conclusions from the following observations:

- The overall ensemble multi-label performances are closely linked to the combination step and an inappropriate use combination of the base-classifier predictions may damage the ensemble predictive performances.
- The ***PC*** strategy is well designed for the instance-wise metrics especially when the base-classifier considers the correlation between labels.
- The ***LC*** strategy is more appropriate for label-wise metrics by locally correcting the ensemble output for each label.
- Multi-Threshold calibration over out-of-bag samples performs well across all multi-label ensemble models and metrics. Given its simplicity and its computational cost, it could be considered as a very simple and practical approach to calibrate the decision threshold toward the objective loss metric.

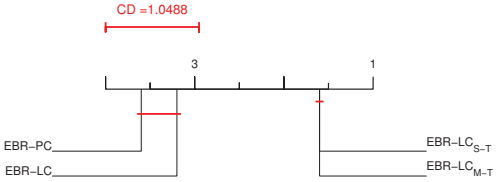
FIGURE 5.1: The critical diagrams for the EBR variants across the six multi-label bi-partition-based metrics: the results from the Nemenyi post-hoc test at 0.05 significance level on the data sets: (a) *Subset 0/1 loss* ; (b) *Jaccard loss* ; (c) *Instance-F1 loss* ; (d) *Micro-F1 loss* ; (e) *Macro-F1 loss* ; (f) *Hamming loss*.



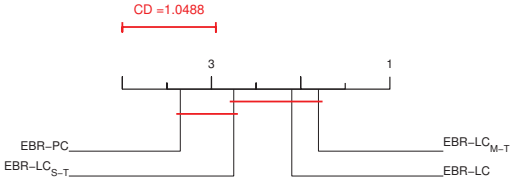
(a) Subset 0/1 loss



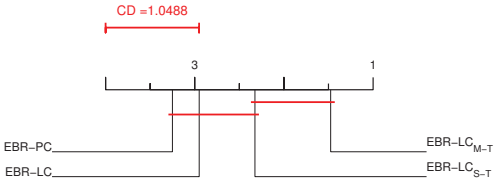
(b) Jaccard loss



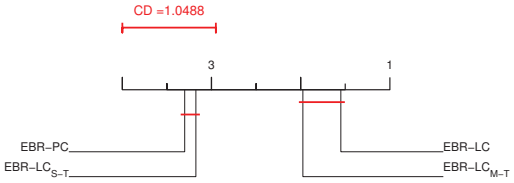
(c) Instance-F1 loss



(d) Micro-F1 loss

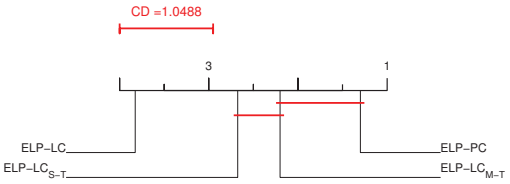


(e) Macro-F1 loss

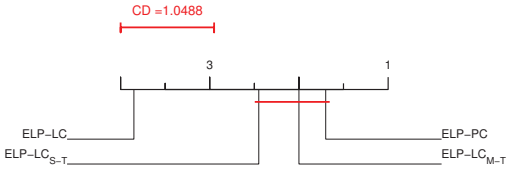


(f) Hamming loss

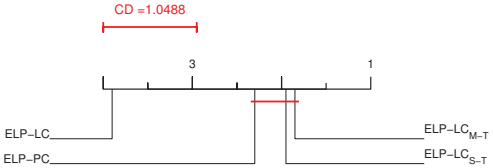
FIGURE 5.2: The critical diagrams for the ELP variants across the six multi-label bi-partition-based metrics: results from the Nemenyi post-hoc test at 0.05 significance level on the data sets: (a) *Subset 0/1 loss* ; (b) *Jaccard loss* ; (c) *Instance-F1 loss* ; (d) *Micro-F1 loss* ; (e) *Macro-F1 loss* ; (f) *Hamming loss*.



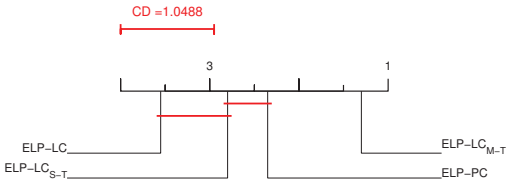
(a) Subset 0/1 loss



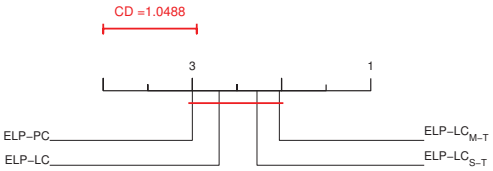
(b) Jaccard loss



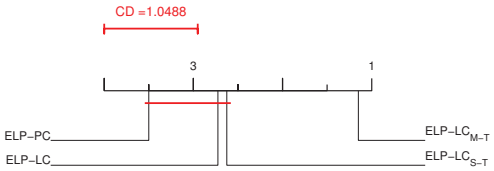
(c) Instance-F1 loss



(d) Micro-F1 loss

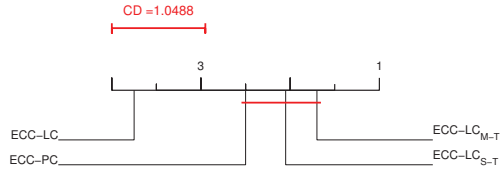


(e) Macro-F1 loss

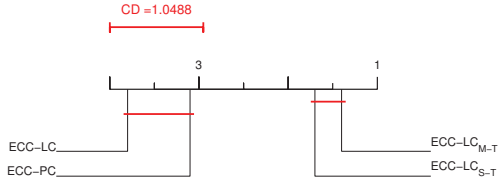


(f) Hamming loss

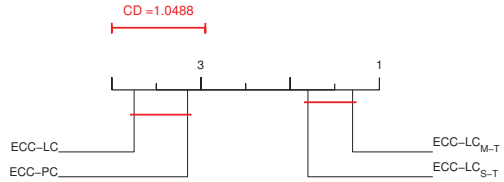
FIGURE 5.3: The critical diagrams for the ECC variants across the six multi-label bi-partition-based metrics: results from the Nemenyi post-hoc test at 0.05 significance level on the data sets: (a) *Subset 0/1 loss* ; (b) *Jaccard loss* ; (c) *Instance-F1 loss* ; (d) *Micro-F1 loss* ; (e) *Macro-F1 loss* ; (f) *Hamming loss*.



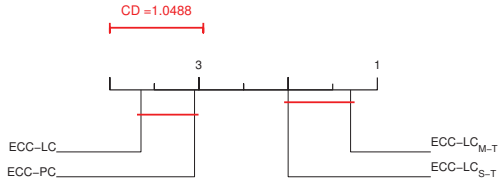
(a) Subset 0/1 loss



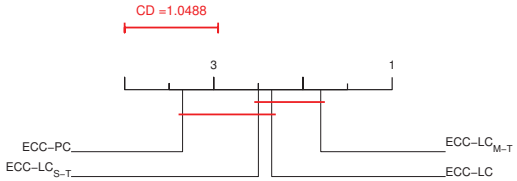
(b) Jaccard loss



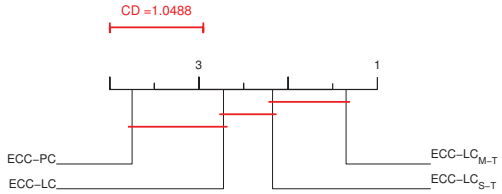
(c) Instance-F1 loss



(d) Micro-F1 loss

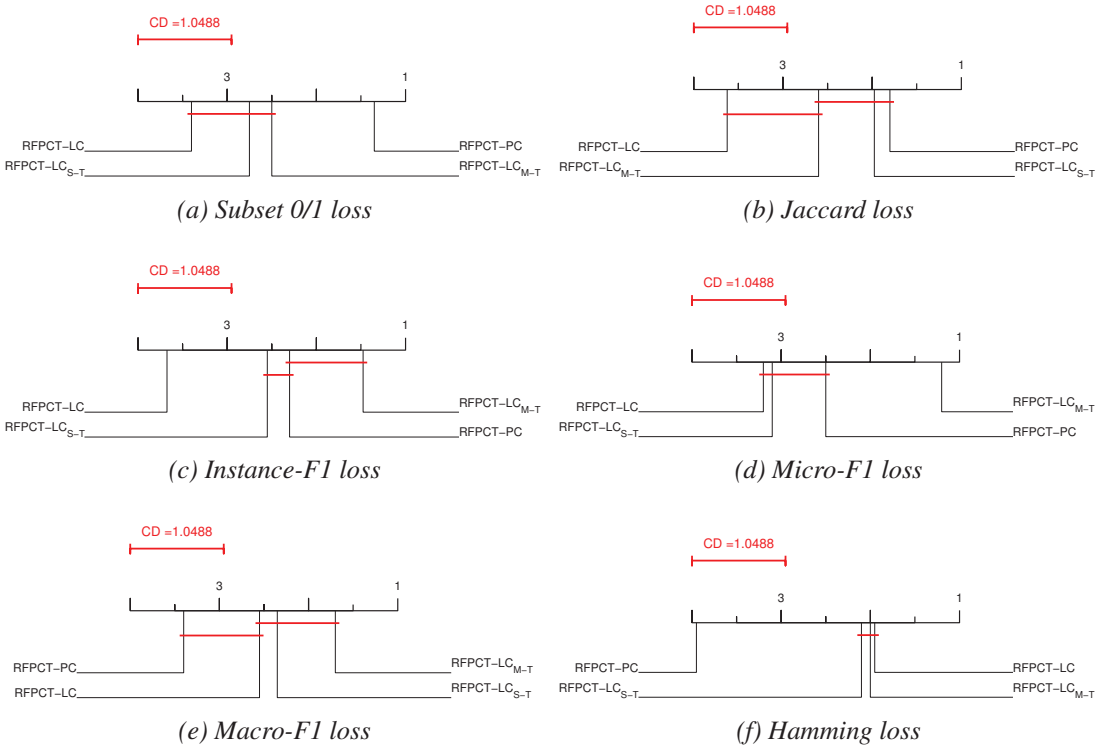


(e) Macro-F1 loss



(f) Hamming loss

FIGURE 5.4: The critical diagrams for the RFPCT variants across the six multi-label bi-partition-based metrics: results from the Nemenyi post-hoc test at 0.05 significance level on the data sets: (a) *Subset 0/1 loss* ; (b) *Jaccard loss* ; (c) *Instance-F1 loss* ; (d) *Micro-F1 loss* ; (e) *Macro-F1 loss* ; (f) *Hamming loss*.

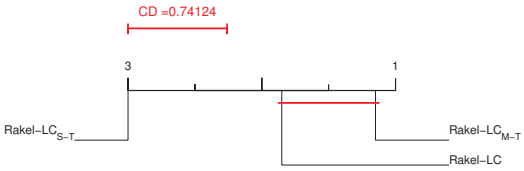


(a) Subset 0/1 loss *(b) Jaccard loss*

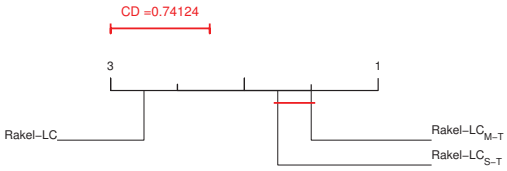
(c) Instance-F1 loss *(d) Micro-F1 loss*

(e) Macro-F1 loss *(f) Hamming loss*

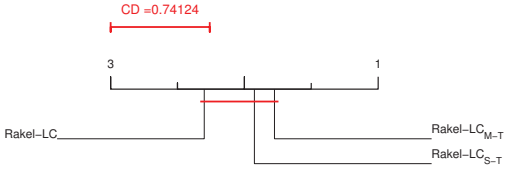
FIGURE 5.5: The critical diagrams for the RAkEL variants across the six multi-label bi-partition-based metrics: results from the Nemenyi post-hoc test at 0.05 significance level on the data sets: (a) *Subset 0/1 loss* ; (b) *Jaccard loss* ; (c) *Instance-F1 loss* ; (d) *Micro-F1 loss* ; (e) *Macro-F1 loss* ; (f) *Hamming loss*.



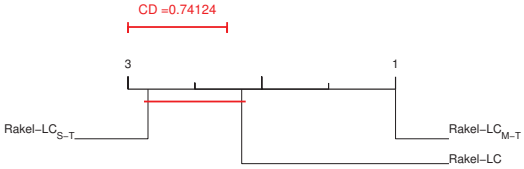
(a) Subset 0/1 loss



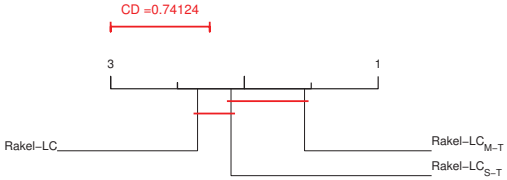
(b) Jaccard loss



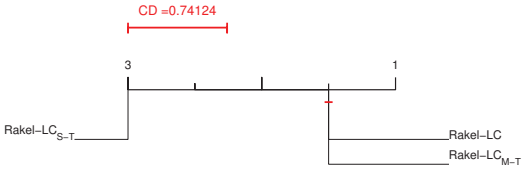
(c) Instance-F1 loss



(d) Micro-F1 loss

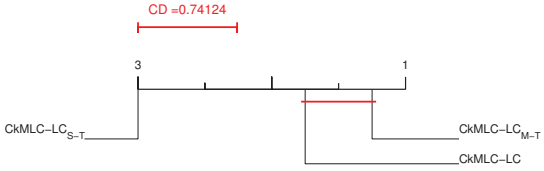


(e) Macro-F1 loss

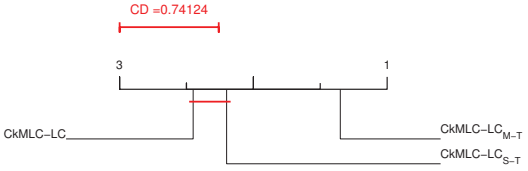


(f) Hamming loss

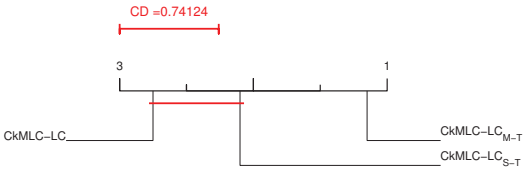
FIGURE 5.6: The critical diagrams for the CkMLC variants across the six multi-label bi-partition-based metrics: results from the Nemenyi post-hoc test at 0.05 significance level on the data sets: (a) *Subset 0/1 loss* ; (b) *Jaccard loss* ; (c) *Instance-F1 loss* ; (d) *Micro-F1 loss* ; (e) *Macro-F1 loss* ; (f) *Hamming loss*.



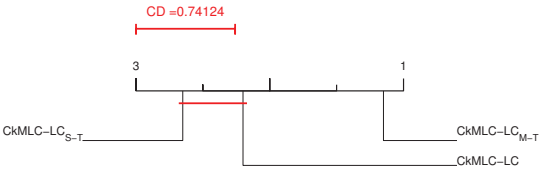
(a) Subset 0/1 loss



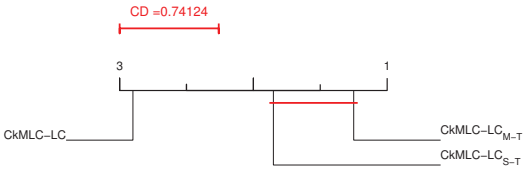
(b) Jaccard loss



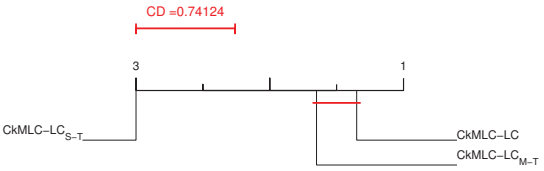
(c) Instance-F1 loss



(d) Micro-F1 loss

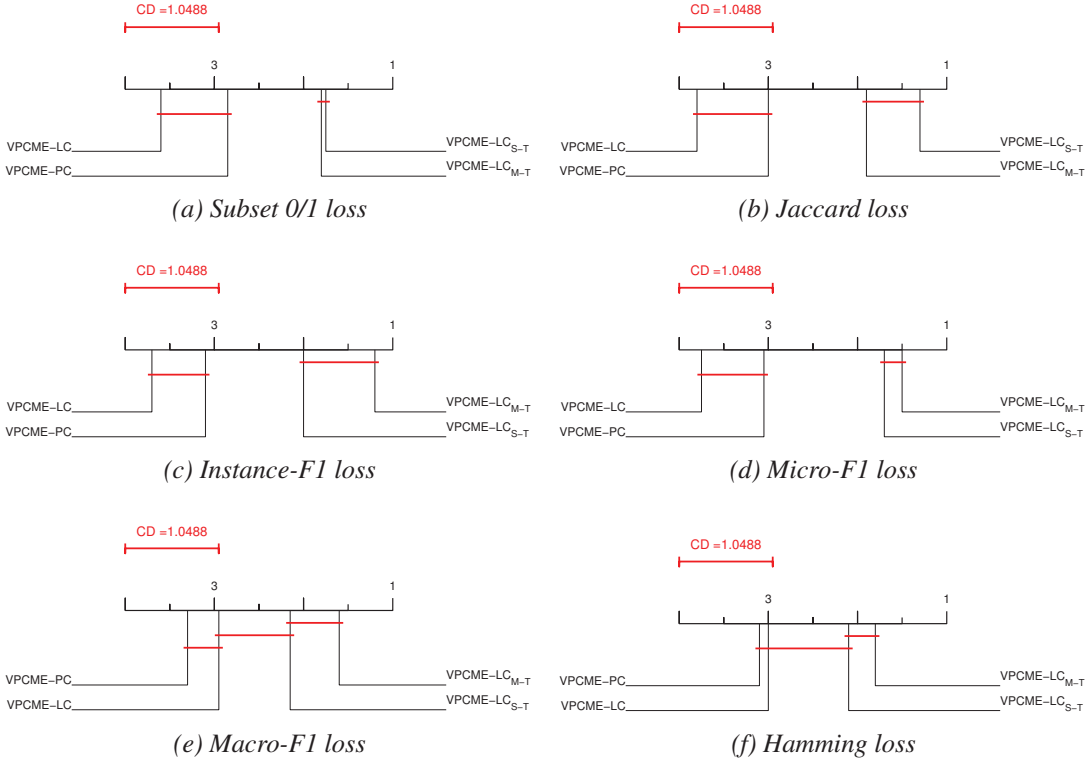


(e) Macro-F1 loss



(f) Hamming loss

FIGURE 5.7: The critical diagrams for the VPCME variants across the six multi-label bi-partition-based metrics: results from the Nemenyi post-hoc test at 0.05 significance level on the data sets: (a) *Subset 0/1 loss* ; (b) *Jaccard loss* ; (c) *Instance-F1 loss* ; (d) *Micro-F1 loss* ; (e) *Macro-F1 loss* ; (f) *Hamming loss*.



(a) Subset 0/1 loss *(b) Jaccard loss*

(c) Instance-F1 loss *(d) Micro-F1 loss*

(e) Macro-F1 loss *(f) Hamming loss*

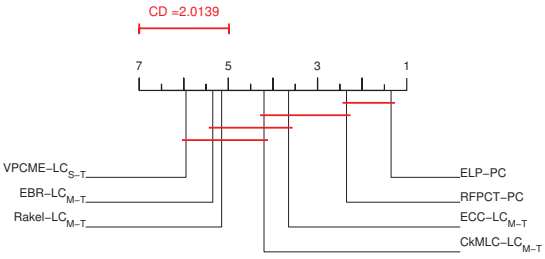
In order to give an overview of the six analyzed models after the combination analysis, we select the best ranked variant of each ensemble model and compare their performances over different metrics using the same the methodology introduced above (Friedman test in tandem with the Nemenyi post-hoc test). It is worth noting that the Multi-threshold variant (*i.e.* $-LC_{M-T}$), that use our *Forward Multi-label Thresholds Calibration* algorithm is the most represented among the best ranked approaches. This validates again the motivation behind our threshold calibration strategy to greatly help ensemble multi-label models to reduce bi-partition-based loss metrics.

In this analysis, the Friedman test reveals statistically significant differences (at $p = 0.05$) between the ensemble approaches across all the metrics. The Nemenyi post-hoc tests with the average rank diagrams are presented on Figure 5.8. The algorithms that do not differ significantly (at $p = 0.05$) are connected with a line. The critical difference CD is shown above the graph (CD=1.9653 here).

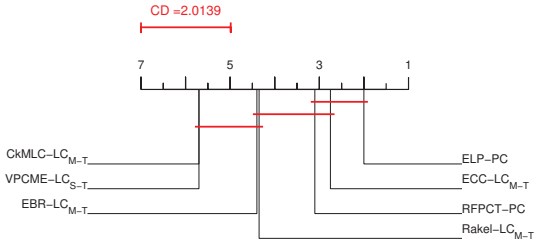
As may be observed in Figure 5.8 the ranks of the models differ *w.r.t.* each metric. However, the ELP and the ECC approaches are generally within the best ranked approaches across all the metrics.

Besides, the RFPCT is located within the best ranked models over the *Subset 0/1 loss*, the *Jaccard loss*, the *Instance-F1 loss* and the *Micro-F1 loss*. By cons, the RFPCT approach is away from the leading group over the label-wise metrics (over the *Macro-F1 loss* and *Hamming loss*). We also note that the VPCME is all usually located in the left side of the diagram within the worst performing group of approaches.

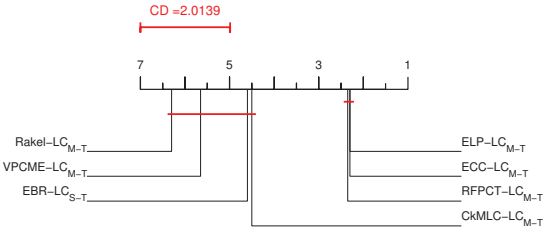
FIGURE 5.8: Average ranks diagrams comparing the six ensemble approaches in terms of (a) *Subset 0/1 loss* ; (b) *Jaccard loss* ; (c) *Instance-F1 loss* ; (d) *Micro-F1 loss* ; (e) *Macro-F1 loss* ; (f) *Hamming loss*



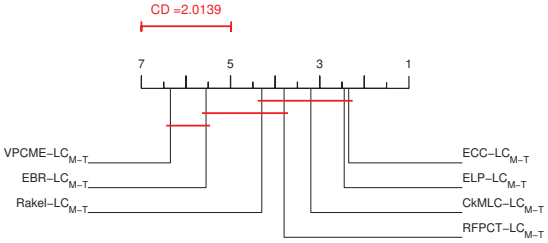
(a) Subset 0/1 loss



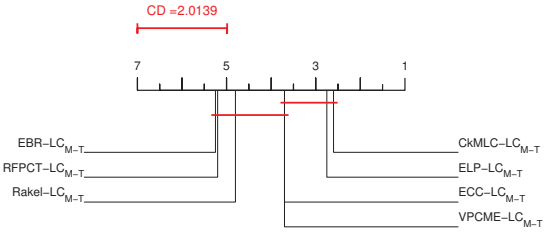
(b) Jaccard loss



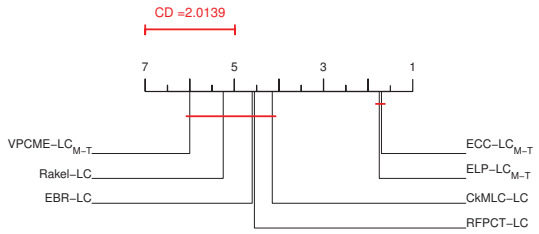
(c) Instance-F1 loss



(d) Micro-F1 loss



(e) Macro-F1 loss



(f) Hamming loss

5.4 Chapter summary

In this Chapter, we addressed the combination strategies in ensemble multi-label models. We proposed, discussed and analyzed two possible combination schemes: *i) The* ***Label-wise Combination strategy*** and *ii) the* ***Powerset-wise Combination strategy***. Then, we investigate the link between the combination strategy and the loss function optimized by the ensemble multi-label model.

Moreover, we discussed the different properties of the proposed strategies and analyzed their behaviour on different ensemble models over different loss metrics. We argued that the combination step should be considered in conjunction with the loss metric on which the predictions will be evaluated as it influences the prediction quality. We corroborated our findings with an extensive empirical analysis over a wide range of multi-label data sets.

Based on our findings, we drew three main conclusions: *i)* For instance-wise performance metrics, it is more appropriate to consider the base-model prediction as an indivisible information by adopting the Powerset-wise Combination strategy. *ii)* The Label-wise Combination strategy is more appropriate for label-wise metrics by locally correcting the ensemble output for each label. *iii)* Multi-Threshold calibration over out-of-bag samples perform well across all multi-label ensemble models for both label-wise and instance-wise metrics.

We believe that these results have some important implications from a methodological and practical point of view. Perhaps one can build an ensemble committee and change the combination step to reach the best-prediction *w.r.t.* different loss functions, using the adequate combination strategy for each metric. Furthermore, given its simplicity and its computational cost, our proposed threshold calibration over out-of-bag samples could be considered as a very practical approach to calibrate the decision threshold to handle efficiently more complex multi-label metrics with an unclear multi-label model from a loss-minimization point of view.

Chapter 6

Feature Selection in Multi-label learning

Similarly to other machine learning tasks, multi-label learning also experiences the curse of dimensionality, which may cause problems when learning from high-dimensional data. Thus, the identification of relevant subsets of random variables -among thousands of potentially irrelevant and redundant variables- is a very important issue to overcome. Multi-label feature selection is an emerging research topic as considerable real-world applications are dealing with high-dimensional data such as text categorization, gene function classification, and semantic annotation of images [92–94]

Unlike single-label feature selection -where the aim is to strike on the most discriminant features for the target label-, in the multi-label context, the feature selection task is more complicated as there is more than one target label. The standard approach for multi-label Feature Selection is to address the task by extending the techniques available for single-label classification via the bridge provided by multi-label transformations.

The multi-label feature selection task becomes more difficult when the amount of labeled data is very limited, in the sense that it is time-consuming or costly to obtain. In such situation, it becomes difficult to build an accurate classification model and more challenging to identify redundant and irrelevant variables from the feature set. In this regard, Semi-supervised multi-label feature selection addresses this problem by using unlabeled data together with labeled data in the feature selection process.

This Chapter focuses on feature selection in supervised and semi-supervised multi-label learning. It will be devoted to present the fundamental concept of feature selection and summarizes the state-of-the-art of proposed feature selection approaches in the supervised and semi-supervised

multi-label contexts. The goal of the Chapter is to provide the necessary background to understand the approaches presented in the following Chapters.

6.1 Features selection : Basic Concepts

As an effective data preprocessing step, feature selection is a vital process to prepare high-dimensional data for numerous data mining and machine learning tasks. Feature selection enables the identification of important features in the data sets. The main goal of the process is to find a subset of features with predictive performance comparable to the full set of features according to an evaluation criterion [95]. The objective is to enable the classification model to achieve good or even better solutions with a restricted subset of features [96]. Thus, providing support to cope with the "curse of dimensionality" problem when learning from high-dimensional data. The feature selection problem is also known as "subset selection" and has been studied by the statistics and machine learning communities for many years. It can efficiently reduce data dimensionality by removing irrelevant and/or redundant features.

Feature selection algorithms use information from labeled data to find the relevant subsets of variables, *i.e.*, those that conjunctively prove useful to construct an efficient classifier from data. By removing irrelevant and/or redundant features, the feature selection process aims to speed up the learning algorithms, better understanding of the underlying process that generates the data, and increase the learning accuracy [96]. Indeed, since the goal -in the supervised setting- is to approximate the underlying function between the input and the output, it is reasonable and inherent to ignore input features with light effect on the output target, to preserve the size of the model small. Various studies show that variables can be removed without performance deterioration [97–100].

From a performance perspective, the aim of the feature selection is to enable the classification model to achieve good or even better solutions with a restricted subset of features [96]. In practice, irrelevant features involved in the learning process may induce significant computational cost and may also lead to over-fitting.

In supervised learning, feature selection approaches are based on a specific feature importance metric to evaluate the feature relevance. Several feature importance measures have been proposed in the literature, such as the Chi-square, ReliefF, Gini Index, Information Gain, Random Forest feature importance [11], to name a few. The feature relevance is evaluated in two main ways: Individual evaluation and subset evaluation. On the one hand, individual evaluation evaluates individual features and assigns them weights (ranks) according to their relevance to the target variable. Thus, the approach is computationally less expensive. Nevertheless, the individual

feature evaluation is inadequate to detect redundant features as they are likely to have similar rankings.

On the other hand, the subset evaluation approach is the brute-force feature selection approach. It can handle both, feature relevance and feature redundancy. Unlike individual evaluation, it exhaustively evaluates all possible combinations of the input variables and then determines the best subset. Obviously, the cost of the exhaustive search approach is prohibitively high, with the considerable risk of over-fitting.

Depending on the interaction with the learning algorithm, features selection methods, are classified in three categories: a) Wrapper methods, b) Embedded methods or c) Filter methods.

In wrapper methods, the learning algorithm output is used to evaluate the importance of features. Each subset of features is evaluated using a measure criterion until finding the best feature set. Wrapper methods have a significant computational cost since they need to evaluate the algorithm's prediction quality for each feature set considered.

As wrapper methods, embedded methods, are related to a learning method. In embedded methods, the relation between the learner and the feature evaluation step is more important than in wrapper method. For the reason that the feature selection process is incorporated in the algorithm's training process, such as decision trees, to decide in each node the feature that has the best ability to discriminate among the target classes.

In filter methods, the feature selection process is conducted independently from the learning algorithm. In those methods, general characteristics of data are used to select the most relevant features. Thus, they have the advantage of being fast and simple to implement. However, unlike the wrapper methods, filters methods may not choose the most suitable features for specific learning algorithms.

Feature selection algorithms based on the embedded and filter approaches may return either a subset of selected features or the weights (measuring feature importance) of all features.

Besides, is also worth mentioning that parallel works on dimension reduction concentrate on feature extraction techniques. Unlike, feature selection techniques -that measure the relevance of individual features (or subsets of features)-, these methods aims to transform the original feature space into a lower dimensional space by proposing new features extracted from the original ones. The new features are built either by using unsupervised or supervised methods such as *Principal Component Analysis*, *Linear Discriminant Analysis*, *Kernel Discriminant Analysis*, to name but a few.

6.2 Supervised multi-label feature selection

Feature selection has been an active research topic in supervised, semi-supervised and unsupervised machine learning, with a large number of related publications and comprehensive surveys [99, 101, 102]. However, most of the works related to supervised feature selection have been mainly to support single-label classification, and less amount of research on multi-label classification have been conducted. This was confirmed by a systematic review process related to multi-label feature selection we carried out in [103].

In multi-label learning, most feature selection tasks have been addressed by extending the techniques available for single-label classification using either the bridge provided by multi-label transformations or adaptation approaches. Most of methods are inspired by the transformation approaches, and propose a previous transformation of multi-label data to single-label data, *i.e.*, to binary data or multi-class data using either the *Binary Relevance* or *Label Powerset* approach.

When the BR strategy is used, it is straightforward to employ a filter approach on each binary classification task, and then combining somehow the results (by averaging for example) [103]. In this context, different feature importance measures have been used, such as Information Gain [97, 104–106], Chi-square [100] and ReliefF [104]. Since each label is treated independently, these methods fail to consider the correlation among different labels. On the other hand, in [107] authors propose the use of ReliefF, which takes into account feature interaction. However, theses methods may not be able to select discriminative features shared by multiple labels.

Methods which perform feature selection considering label correlation are based instead over the *Label Powerset* transformation approach. The Chi-square measure is applied after a Label Powerset transformation in [100]. In [108] an evaluation measure which concerns the ranking quality between output labels is used. The Mutual Information measure is applied in [109] according to a Pruned Powerset Transformation (PPT) [16], which also considers the abel dependence in the feature selection process. The proposed approach termed PPT-MI uses the Pruned Problem Transformation to avoid the Label Powerset transformation drawbacks, then applies a sequential forward selection with the Mutual Information as a search criterion. The Symmetrical Uncertainty measure is extended in [110] to find relationships between all pairs of features and labels.

However, these methods fail in two extreme cases where *(1)* the feature selection model may completely ignore any links or correlation within the labels by considering each label separately [104], or on the contrary, *(2)* it considers each label combination as a meta-class, in an LP style feature selection model [104, 109].

As a first attempt for a multi-label feature selection model that takes into account the label interactions without resorting to problem transformation, Lee and Kim proposed the PMU approach [111]. PMU is a filter multivariate mutual information feature selection that naturally derives

from mutual information between a set of features and a set of labels. It evaluates the feature importance by considering jointly correlation between labels and variables.

In [108, 112], the wrapper approach is directly addressed in multi-label data using evaluation measures and a meta-heuristic to search for the best feature subset, while embedded feature selection based on decision tree classifiers are suggested in [6, 113].

In contrast to these previous filter approaches, Gu el al. propose an embedded-style feature selection method for multi-label learning called CMLFS [114]. CMLFS (for Correlated Multi-Label Feature Selection) is based on LaRank SVM, which is among state-of-the-art multi-label learning methods. In the proposed method, the goal is to find a subset of features, based on which the label correlation regularized loss of label ranking is minimized. Although this method considers correlation among labels, it optimizes a set of parameters during feature selection process to tune the kernel function of multi-label classifier making it impractical in the viewpoint of computational cost [111].

6.3 Semi-Supervised multi-label feature selection

In many real-world applications, the amount of labeled data is very limited, in the sense that it is time-consuming or extremely expensive to obtain. In such situation, there are mainly two challenges. First, in the presence of few amount of labeled data, it becomes difficult to build an accurate multi-label model. And even more to conduct feature selection, since that traditional feature selection algorithm use information from labeled data to find the relevant subsets of variables. Meanwhile a large amount of unlabeled data may be relatively easy to collect, but there has been few ways to use them. Semi-supervised multi-label learning addresses this problem by using unlabeled data together with multi-labeled data in the training process, to enhance the performance of the learned classifiers. On the other hand, the labels in multi-labeled data are typically interdependent and correlated, which poses more difficulties to identify or remove redundant and irrelevant variables from the feature set, especially in high-dimensional data. To overcome this problem, feature selection methods need to explicitly model the label interactions in evaluating the quality of features, which is crucial for better performance.

In the following we will review the semi-supervised multi-label classification and semi-supervised multi-label feature selection approaches that appeared in the literature.

6.3.1 Semi-supervised multi-label classification

Semi-supervised multi-label approaches are proposed to deal simultaneously with few labeled instances and a large amount of unlabeled instances while getting benefit from the information provided by unlabelled data.

In such learning configuration, the key assumption is that two examples will be assigned to similar labels if they overlap in their input space. In [115], Liu *et. al.* formulate the semi-supervised multi-label task as a Constrained Non-negative Matrix Factorization problem, where the objective is to minimize the difference between the instance similarity matrix of the feature space and the similarity matrix of the label space to determine the labels of unlabeled data.

Besides, Chen *et. al.* [116] propose a semi-supervised approach based on two graphs of similarities. The first corresponds to the instance level, with nodes and edges representing respectively instances and pairwise similarities between instances meanwhile, the second graph corresponds to the label level. The idea is to combine the regularization terms for the two graphs (i.e. instance graph and label graph) in a regularization framework where the labels of unlabeled instances were obtained by solving a Sylvester Equation. In an another graph-base approach, Wong *et. al.* [117] present an effective multi-label classification algorithm that simultaneously models the labeling consistency between similar videos and the multi-label interdependence for each video. The model is based on a discrete hidden Markov random field approach for transductive multi-label classification which preserves the multi-label co-positive, co-negative and mutual-exclusive interdependence over the unlabeled and the labeled data points. In [118], *Guo* and *Schuurmans* propose another transductive algorithm which exploits unlabeled data to learn simultaneously the underlying subspace feature representations of the data with a large margin multi-label classification model. Zha *et al.* [119] proposed other graph-based framework which uses one loss function and two types of regularizers. The first is adopted to handle the label consistency on the graph while the second is used to tackle the correlations of multiple labels. Based on this framework, two graph-based algorithms were developed. The idea is to learn the cardinality of the labeled instance to assign new label sets to unlabeled instances using the estimated label concept compositions.

Most of the works on semi-supervised multi-label learning are graph-based approaches and differ only in the way that regularization term affects the labels and the features. These methods work only in transductive setting and require that all unlabeled instances to be available during training, since that the learned classifier can only predict the labels of unlabeled data used during training, and can not generalize to unseen new test instances. Nonetheless, it is worth citing a recent different approach, named iMLCU for inductive Multi-Label Classification with Unlabeled data [120], which tackles semi-supervised multi-label learning under the inductive setting by adapting the semi-supervised support vector machines. The semi-supervised multi-label classification

task is formulated as an optimization problem of q linear models that fits the labeled instances by exploiting pairwise label correlations and uses the unlabeled instances for regularization. The resulting optimization problem of empirical loss term on labeled data and regularization term on unlabeled data, which is non-convex and solved via the ConCave Convex Procedure [121]. However, these proposed methods simply explore the multi-label inter-similarity and impose the smoothness assumption of the labels over each data point which is not accordant in practice, since that excludes the mutual–exclusive links within the labels [117]. Moreover, their formulation leads to complex optimization problems for which the computational cost is very expensive.

More recently, a new approach named Coins, for *CO-training for INductive Semi-supervised multi-label learning* is proposed [122]. The approach adapt the co-training strategy in the multi-label context. In each co-training round, a dichotomy over the input feature space is learned by maximizing the diversity between the two classifiers. Then, pairwise ranking predictions on unlabeled data are communicated between either classifier for the model refinement.

6.3.2 Semi-supervised multi-label feature selection algorithms

In the multi-label context, the feature selection task is considered as a more difficult problem as there is more than one target label. And in a multi-label semi-supervised setting, the task becomes more challenging. Although considerable attention has been given recently to multi-label feature selection where different sophisticated approaches have been proposed, little attention has been given to consider feature selection in the semi-supervised multi-label setting. Existing multi-label feature selection algorithms are designed for the supervised setting. They need a sufficient amount of labeled training data and are not able to handle both labeled and unlabeled data.

The key for designing an effective semi-supervised multi-label feature selection algorithm is to develop a framework, under which the relevance of a feature can be evaluated by both labeled and unlabeled data in a natural way.

Recently, Chang *et al.* proposed a convex semi-supervised multi-label feature selection algorithm for large-scale multimedia analysis, named (CSFS) for *Convex Semi-supervised multi-label Feature Selection* [123]. The proposed algorithm makes use of both labeled and unlabeled instances to select feature while taking into account correlation within the labels. Besides, Alalga *et al.* [124] proposed a scoring function for measuring the relevance of each feature called S-CLS for *soft-constrained Laplacian score*. The proposed scoring framework is based on the Laplacian score and reflects the correlation of the feature to the label.

More recently, a semi-supervised multi-label feature selection method leveraging shared information among multiple labels is proposed [125]. The method is based on graph matrix formulation of the semi-supervised multi-label task to model the geometric structure of the training

data over both labeled and unlabeled examples. It uses a l_1-norm based graph matrix is imposed to capture a clear underlying manifold structure in the multi-label target space. To select the representative features, the model considers the shared subspace learning approach and uses a l_2-norm to select the most representative features. An iterative algorithm is proposed to optimize the non-smooth objective function, involving the both l_2-norm and l_1-norm. The proposed algorithm has only been applied for three different applications: natural scene classification, web page annotation, and yeast gene functional classification.

It is also worth mentioning that there are parallel works for dimension reduction in the semi-supervised learning, which uses semi-supervised multi-label data to achieve efficient dimensionality reduction [126]. This category of methods have demonstrated their effectiveness in various application domains such as image annotation [126], but unfortunately, their detailed description is beyond the scope of this thesis.

6.4 Chapter summary

Multi-label feature selection is an active area of research today, with more recent proposals. This Chapter introduced the multi-label feature selection and overviewed the proposed multi-label feature selection techniques in both supervised anD semi-supervised ways.

The Chapter first gives the basic concept of the feature selection and terminology. It then presents the proposed works in the supervised multi-label feature selection.

As discussed the transformation approach are the most popular strategy in the proposed multi-label feature selection methods. This could be explained the advantage given by the transformation approach to apply existing single-label feature selection method. Further-more this choice is often coupled with filter approaches which is partly justified by the relative lower computational cost in comparison with other alternatives. Only a few works adopt a multi-label perspective in term of metrics to handle the feature selection task.

The Chapter also presented the semi-supervised multi-label learning and overviewed proposed classification algorithm and feature selection methods.

Most of the works on semi-supervised multi-label learning are graph-based approaches. Generally, graph-based semi-supervised techniques are utilized to construct an affinity matrix over the labeled and unlabeled data. Then the classifications of unlabeled data are obtained via label propagation. Furthermore, these propositions work under the transductive setting, which only focus on classifying given unlabeled data and thus cannot generalize to unseen instances. Besides, although considerable attention has been given recently to multi-label feature selection

where different sophisticated approaches have been proposed, little attention has been given to consider feature selection in the semi-supervised multi-label setting.

Under this overview, we observed that little attention has been given to exploiting the power of ensemble methods with a view to identify and remove the irrelevant features in a multi-label setting. Such methods are shown to be very beneficial for enhancing the robustness and the generalization ability of single learners and overcoming the curse of dimensionality problem. Ensemble methods, in particular Random Forest [127] have been proved to be effective for estimating feature importance in traditional single-label [127], semi-supervised [128] and unsupervised [129–131] learning. Therefore, in the Chapter 7 we naturally adapt the traditional Random Forest permutation importance measure to the multi-label scenario via three different strategies. Then in Chapter 8 we extend our proposed ensemble model CkMLC to the semi-supervised context. The proposed approach combines ideas from co-training and random k-labelsets ensemble learning with a new permutation-based out-of-bag feature importance measure.

Chapter 7

Multi-Label Feature Selection Using the Random Forest Paradigm

The identification of relevant subsets of random variables, among thousands of potentially irrelevant and redundant variables, is a very important topic of pattern recognition research that has attracted much attention over the last few years.

As aforementioned in Chapter 6, multi-label feature selection has been widely studied and have encountered some success in many applications during the past few years [104, 111, 114]. However, little attention has been given to exploiting the power of ensemble methods with a view to identify and remove the irrelevant features in a multi-label setting. Such methods which combines multiple base learners to jointly accomplish one common task are shown to be very beneficial for enhancing the robustness and the generalization ability of single learners and overcoming the curse of dimensionality problem. Besides, ensemble methods, in particular Random Forest (RF) [127], which originally inspired this work, have been proved to be effective for estimating feature importance in traditional single-label [127], semi-supervised [128] and unsupervised [129–131] learning. On the other hand, the diversity of multi-label classification evaluation performances create confusion towards the classification algorithm effectiveness; and even more towards multi-label feature selection relevance.

Motivated by this, we discuss in the sequel, three different wrapper multi-label feature selection strategies [132] based on Random Forest paradigm. These variants optimize different loss functions depending on the way label dependence is operated. We also analyze how the optimized loss function in the multi-label classifier influences the relevance of a multi-label feature selection process, thereby contributing to a better understanding of the internal meaning of selected features.

The main contributions of this work are highlighted as follows :

- In multi-label classification task, authors in [133] showed, on the basis of theoretical and empirical results, that there is a strong connection between the optimized performance measure and the way the dependencies between class labels are modeled. In this regard, we believe that the type of loss function has a strong influence on whether or not an exploitation of label dependencies can be expected to yield a true benefit for feature selection results. Perhaps most importantly, it cannot be expected that the same multi-label feature selection method to be optimal for different types of losses at the same time. The main proposal of this Chapter is grounded on this consideration. We pursue this direction to elaborate more closely on the idea of exploiting label dependence, thereby contributing to a better understanding of multi- label feature selection.

- We discuss three wrapper multi-label feature selection methods [132], which use the RF paradigm. The three RF-based approaches differ in their considerations of label dependence and its connection with the optimized loss function. Differences between these approaches lead to different feature selections each one adapted to optimize specific loss function during the RF feature selection process. The three RF variants called BRRF, RFLP and RFPCT, stand respectively for BRRF, for *Binary Relevance Random Forest* and RFLP, for *Random Forest Label power-Set*, consists of the two problem transformation approaches BR and LP, to previously transform the multi-label data into single-label data, which is then used to perform a Random Forest. However, RFPCT [134] (Random Forest of Predictive Clustering Trees) is another extension of RF that uses as base classifier PCT [135], a decision tree predicting multiple target attributes at once. We would like to mention that feature selection using RFPCT was initially proposed in [136], nonetheless, it was evaluated on a single biological data set and only compared to a trivial random feature ranking algorithm in [137].

- Extensive experimental comparison were conducted on 13 various real-life multi-labeled data sets to evaluate the power of RF-based multi-label feature selection methods. Results support the main claims of this work concerning loss minimization and its relationship with label dependence consideration in the multi-label feature selection process. They also demonstrate that RF handles accurately the feature selection in multi- label context and enjoys significant advantages compared to other recently proposed methods.

In the remaining of this Chapter, we first study the three RF-based multi-label feature selection methods and describe how variable importance used in RF can be extended in multi-label context. Then, we present our experimental study using real-life multi-label data sets to confront these strategies against recently proposed multi-label feature selection approaches.

7.1 Random Forest-based multi-label feature selection

RF has several desirable characteristics for feature selection: It is robust, exhibits high-quality predictive performance, does not overfit and handles simultaneously categorical and continuous features [127]. Furthermore, RF have proved to be efficient in traditional supervised [127], semi-supervised [128], and unsupervised [131] feature selection process. This section introduces three wrapper multi-label feature selection methods, which use the RF paradigm. In this way, we discuss three variants of RF for Multi-label learning *Random forest of predictive clustering trees* (RFPCT), *Binary Relevance Random Forest* (BRRF), and *Random Forest Label Power-set* (RFLP); and then exploit the *RF permutation importance measure* [127] to evaluate the goodness of a feature. Before introducing the proposed methods, we recall how RF with permutation based out-of-bag (oob) measures feature importance.

The variable importance measure in RF is based on the decrease of predictive performance when values of a descriptive variable in a node of a tree are permuted randomly. Basically, a bootstrap is used as training set to create trees in the forest. In each bootstrapped data set, almost 33% are left oob, *i.e.*, they are not used for the construction of the t^{th} corresponding model h_t ($t \in \{1, \ldots, T\}$). We refer to them as Oob_t. Thus, these instances can be used to estimate non biased feature relevancies. In every tree grown in the forest, the values of the f^{th} feature in the Oob_t data, is randomly permuted to form Oob_t^f, and the tree h_t is used to predict the labels of the new oob patterns. The predictive performance of each tree h_t is evaluated on the untouched oob data and the permuted versions of the oob data. The importance of the f^{th} variable is then calculated as the relative increase of the error that is obtained when its values are randomly permuted (*c.f.* Equation 7.1). The average of this number over all trees in the forest is the importance score for variable f. We note that the greater the value of the importance measure, the more relevant is the feature. A formal description of the pseudocode is given in Algorithm 4.

$$I^f = \frac{1}{T} \sum_{t=1}^{T} \frac{e(h_t(Oob_t^f)) - e(h_t(Oob_t))}{e(h_t(Oob_t))} \tag{7.1}$$

where T is the size of the forest and e is the error measure function.

Given a label space $\mathcal{L} = \{\lambda_1, \lambda_2, ..., \lambda_q\}$ and a data set $\mathcal{D}$ that consists of n instances each taking the form $(\mathbf{x}_i, \mathbf{y}_i)$ where $\mathbf{x}_i = (x_i^1, \cdots, x_i^M)$ is a vector of M descriptive features and $\mathbf{y}_i \in \mathcal{L}$ is the subset of labels associated to $\mathbf{x}_i$ (represented by a binary feature vector $(y_i^1, y_i^2, \cdots, y_i^q) \in \{0, 1\}^q$), we present, in the sequel, the three used variants of RF for multi-label learning and describe how variable importance used in RF can be extended in this context.

Algorithm 4 *Feature importance estimation using Oob*

Require:

D : samples database;

M : feature space cardinality

T : forest size;

h_t : tree learning algorithm

1: $I = 0$
2: **for** $t \in \{1, \ldots, T\}$ **do**
3: $Bag_t \leftarrow$ bootstrap sample from D
4: $Oob_t \leftarrow E \backslash Bag_t$
5: $h_t \leftarrow learn\ a\ tree\ from\ Bag_t$
6: **for** $f \in \{1, \ldots, M\}$ **do**
7: $Oob_t^f \leftarrow Randomize(Oob_t, f)$
8: $I^f \leftarrow I^f + \frac{1}{T} . \frac{e(h_t(Oob_t^f)) - e(h_t(Oob_t))}{e(h_t(Oob_t))}$
9: **end for**
10: **end for**
11: **return** I

7.1.1 Binary Relevance Random Forest (BRRF)

This method transforms the multi-label data set $\mathcal{D}$ into many single-label data sets, one for each individual label in $\lambda_i \in \mathcal{L}$. After this transformation, a RF is created for each label λ_i. The relevance of each feature according to each individual label is measured using the above Equation 7.1 for which e is the traditional single-label classification error. Finally, the average of the score of all features across all labels is considered. BRRF, focuses on each label individually and does not take into account label dependence. Consequently, it gives a local feature selection. Note that in [133], a concrete connection between the type of multi-label classifier used and the loss to be minimized has been established, showing that BR is optimal for decomposable loss functions over labels, such as *Hamming loss*.

7.1.2 Random Forest Label Power-set (RFLP)

In this method the multi-label feature selection problem is handled using the Label Powerset (LP) strategy. This approach reduces the multi-label data set $\mathcal{D}$ to a multi-class data set by treating each distinct labelset as an unique multi-class label. To avoid creating too many rarely classes, causing overfitting and imbalance problems the Pruned Problem Transformation in [109] was used; patterns with too rarely occurring labels are simply removed from the training set by

considering labelsets with a predefined minimum occurrence. A RF could be now performed and the above described feature selection procedure will be naturally applied using in Equation 7.1 the traditional single-label classification error e. In this way, this approach directly takes into account label correlation. It is worth noting that, according to theoretical claims in [133], LP should perform well for the *subset 0/1 loss* metric.

7.1.3 Random Forest Predictive Clustering Tree (RFPCT)

In contrast to both previous approaches (BRRF and RFLP) for which the RF grows many classification trees using a CART as a base classifier, RFPCT [134] is an extension of RF that use a randomized variant of the non Pruned Predictive Clustering Tree (PCT) [135], as a base classifier. In this approach, the multi-label data $\mathcal{D}$ is handled directly and is then able to provide an intuitive way for taking into account relationships between labels. Nevertheless, it is noteworthy that BRRF and RFPCT perform comparably for classification (see [134] for more details).

The feature selection problem with RFPCT follows the same procedure described above. Feature relevances are measured on each PCT tree, and then averaged over all the trees in the forest. However, since PCT is an adaptation method devoted to learning simultaneously all the labels, the RF-based feature evaluation procedure requires an appropriate multi-label error measure e instead of the ordinary classification error used for BRRF and RFLP. As suggested in [136, 137], the multi-label error for each tree in the forest is obtained by averaging the individual classification errors across the L labels. It is worth remarking though that this error was defined independently of the model-performance metric, here the global accuracy.

7.1.4 Computational complexity

In this section, we analyze and discuss the computational complexity aspects of the three RF-based multi-label feature selection methods. For this purpose, we identify two phases: in the first phase a random forest is built, in the second phase the structure of the forest is used to generate feature importance.

In BRRF a random forest is constructed for each label in $\mathcal{L} = \{\lambda_1, \lambda_2, ..., \lambda_q\}$. In each forest the computational complexity of inducing a random tree scales as $O(an\log(n))$ where a denotes the number of tests considered to construct a node ($a = f(M)$ in our case, where M is the number of features) and n stands for the number of elements in the data set, under the assumption that a reasonably symmetric tree is built (the depth of which is logarithmic in the number of leaves) and that the evaluation of a single test takes constant time in the size of the data set (see [138] for more details). The complexity for the first phase, the induction of the whole q random forest, scales then as $O(qTMn\log(n))$, where T is the size of each forest. The complexity of the second

phase in BRRF (RF permutation feature importance measure) depends on the prediction costs with a decision tree and the random permutation of descriptive attributes in the *Oob* data. In every tree of a forest of a given label, each feature f from the M descriptive ones is shuffled (randomly permuted) in the *Oob* cases ($O(n)$). These *Oob* instances of size n are then re-classified in $O(n\log(n))$ steps. The importance of variable f is then measured as the relative increase of the single-label error in the *Oob* permuted instances ($O(n)$). The dominant term for measuring importance for the feature f in every tree is $O(n\log(n))$. Hence, measuring variable importance for all M descriptive variables using all T trees in a forest of a given label costs $O(TMn\log(n))$. Consequently, the complexity of the second phase overall the L labels is $O(qTMn\log(n))$, which means that BRRF takes order $O(qTMn\log(n))$ steps. Note that BRRF can easily be parallelized.

The derivation of the computational complexity of RFPCT for feature importance evaluation is very similar. In RFPCT, the computational complexity of inducing a PCT tree scales as $O(aqn\log(n))$ with $a = f(M)$. The difference here lies in the procedure for calculating the best split at a given node. This procedure, now scales as $O(aqn)$ instead of $O(an)$. So, the overall computational complexity of constructing a random forest of PCT is $O(qTMn\log(n))$. In the second phase of RFPCT and in every tree of the forest, each feature (out of M) is randomly permuted in the *Oob* cases ($O(n)$). These *Oob* instances of size n are then classified again in $O(n\log(n))$. The importance of variable f is then measured as the relative increase of the multi-label error in the *Oob* permuted instances ($O(qn)$). For each feature, it takes $O(n\log(n) + qn)$. Consequently, for the M features and T trees in RFPCT, it scales as $O(TMn\log(n) + qTMn)$. This means that the computational complexity of RFPCT is dominated by the random forest construction ($O(qTMn\log(n))$), as observed with BRRF.

In RFLP the multi-label data set is first transformed into one single-label data set in $O(n\log(n))$ and then a random forest is constructed in $O(TMn\log(n))$. Bearing in mind that the second phase in RFLP follows the same scheme as in RFPCT, the overall complexity of RFLP is $O(TMn\log(n)+TMnq)$. Let us assume that $q < \log(n)$. This means that the dominant term in the computational complexity of RFLP is $O(TMn\log(n))$. Considering this, RFLP reduces the computational complexity by a factor $O(q)$ compared to BRRF and RFPCT. On the other hand, if we assume that $q > \log(n)$, the dominant term is equal to $O(TMnq)$. In this case, RFLP reduces the computational complexity by a factor $O(\log(n))$.

7.2 Performances analysis

This section presents an experimental study using benchmark data to confront the different variants of feature selection models. We investigate the effectiveness of the RF-based feature importance measures for multi-label feature selection regarding the optimized loss function; and compared their performances against recently proposed multi-label feature selection methods.

TABLE 7.1: Description of the Benchmark multi-label data sets used in the experiments.

Data set	Domain	q	M	Training set		Test Set	
				N	Card	N	Card
Arts	Text	26	462	2000	1.627	3000	1.642
Business	Text	30	438	2000	1.590	3000	1.586
Education	Text	33	550	2000	1.465	3000	1.458
Emotions	Music	6	72	391	1.813	202	1.975
Enron	Text	53	1001	1123	3.387	579	3.363
Entertainment	Text	21	640	2000	1.426	3000	1.417
Health	Text	32	612	2000	1.667	3000	1.659
Medical	Text	45	1449	333	1.255	645	1.240
Scene	Image	6	294	1211	1.062	199	1.086
Science	Text	40	743	2000	1.489	3000	1.425
Slashdot	Text	22	1079	1513	1.174	2269	1.185
Social	Text	39	1047	2000	1.274	3000	1.290
Yeast	Biology	14	103	1500	4.228	917	4.252

7.2.1 Data sets and evaluation protocol

To confront the different variants of feature selection, we use 13 benchmark multi-label data sets obtained from the *Mulan's repository* [87]. The selected data sets were used in various studies and evaluations of multi-label learning methods. It covers different application domains: Biology, semantic scene analysis, music emotions and text categorization. From the literature, these data sets come pre-divided into training and testing parts; thus, in the experiments, we use the original training and test sets in their original format. This also allow an easier comparison to future and already published studies.

Table 7.1 summarizes basic statistics of the data sets: the number of features (**M**); the number of labels (**q**) and the Label Cardinality (**Card**), which is the average number of single labels associated with each instance.

We confronted the three variants of RF-based multi-label feature selection methods to two recently proposed ones: PPT-MI [109] and PMU [111]. PPT-MI is a multi-label feature selection method using the Pruned Problem Transformation (PPT) to improve the LP approach followed by a sequential forward selection with the Mutual information (MI) as search criterion. PMU is a filter approach that takes into account label interactions in evaluating the dependency of given features without resorting to problem transformation. It is presented as a multivariate mutual information-based feature selection method for multi-label learning that naturally derives from

mutual information between selected features and a set of labels. Guided by considering jointly correlation between labels and variables, both approaches (PPT-MI and PMU), seek to minimize the joint conditional distribution error. We also compared these approaches to a Binary relevance feature selection strategy using mutual information. Such as BRRF, this feature selection gives a feature raking for each label. We denote this approach by BRMI for Binary Relevance Mutual Information. For PMU, BRMI and PPT-MI, the numeric data sets are discretized using the Equal-width interval scheme, as suggested by the authors in [111]. Furthermore, the three variants of RF of multi-label learning (BRRF, RFLP and RFPCT) are tuned similarly. The number of variables to split on at each node and the committee size are set to $\sqrt{M}$, and 100, respectively.

To evaluate the predictive performance of the compared multi-label feature selection algorithms, we used two multi-label classification schemes: Binary relevance scheme, where each label is treated independently and does not take into account dependencies among labels. This scheme is favorable to boost the performance of multi-label loss functions with marginal conditional distributions as *Hamming loss* [133]. Label Power Set scheme, where correlation between labels is taken into consideration. This scheme improve the performance of loss functions that estimate the joint conditional distribution as the *Subset 0/1 loss* [133]. Both multi label classification scheme were instantiated with the LIBSVM (with linear kernel).

As mentioned above, BRRF and BRMI generate, for each label, a specific feature ranking. This leads specific feature pertinence for each label. For BR scheme, this property is operable by allowing each classifier to focus on most discriminative features for each single label. For LP scheme, specific label feature importance is aggregated by averaging features importance (or features ranking) across all labels to generate a common feature label raking for all labels. Although, RFLP, RFPCT, PMU and PPT-MI, generate a single ordered common list of features toward all labels which convenient for both strategies where the classifiers, in BR strategy, learn from the same relevant features.

In order to better assess the results obtained for each feature selection algorithm and following the risk minimized by each scheme (BR and LP), we restricted the evaluation measures used in this experiment on two performance measures: *Hamming loss* and *Subset 0/1 loss*.

7.2.2 Comparison results

In the sequel, we present the results obtained from our empirical study and concludes on the applicability and performance of RF for multi-label feature selection.

Tables 7.2 and 7.3 reports the averaged results of the six feature selection methods over the top 50 features (as used in [111]) obtained with both BR and LP schemas for respectively *Hamming*

TABLE 7.2: *Hamming loss* of all feature selection approaches and all data sets using BR and LP as base multi-label learning algorithm. Bold cells highlight the best performing algorithms for each data set.

Data set	ML Base learner	BRRF	BRMI	RFLP	RFPCT	PMU	PPT-MI
Arts	BR	**.0559 ± .001**	.0558 ± .002	.0577 ± .002	.0575 ± .001	.0603 ± .001	.0580 ± .002
	LP	.0769 ± .002	.0748 ± .003	.0722 ± .004	.0728 ± .003	.0768 ± .002	.0738 ± .003
Business	BR	**.0268 ± .001**	.0271 ± .001	.0273 ± .001	.0282 ± .001	.0284 ± .001	.0280 ± .004
	LP	.0284 ± .001	.0286 ± .001	.0274 ± .001	.0283 ± .001	.0283 ± .001	.0275 ± .001
Education	BR	**.0393 ± .001**	.0398 ± .001	.0412 ± .001	.0412 ± .001	.0413 ± .001	.0407 ± .001
	LP	.0529 ± .001	.0508 ± .002	.0499 ± .002	.0508 ± .001	.0494 ± .001	.0489 ± .001
Emotions	BR	**.2340 ± .015**	.2477 ± .017	.2383 ± .015	.2373 ± .012	.2657 ± .029	.2452 ± .024
	LP	.2552 ± .047	.2483 ± .034	.2504 ± .021	.2498 ± .015	.3062 ± .063	.2552 ± .033
Enron	BR	**.0486 ± .002**	.0507 ± .002	.0517 ± .003	.0532 ± .002	.0527 ± .002	.0535 ± .002
	LP	.0610 ± .002	.0655 ± .001	.0608 ± .001	.0610 ± .001	.0611 ± .002	.0604 ± .001
Entertainment	BR	**.0549 ± .001**	.0564 ± .006	.0591 ± .003	.0590 ± .003	.0655 ± .001	.0594 ± .003
	LP	.0813 ± .003	.0781 ± .004	.0761 ± .005	.0775 ± .004	.0829 ± .002	.0770 ± .004
Health	BR	**.0365 ± .003**	.0371 ± .003	.0413 ± .003	.0405 ± .003	.0431 ± .002	.0404 ± .002
	LP	.0496 ± .001	.0489 ± .002	.0443 ± .003	.0429 ± .001	.0453 ± .001	.0429 ± .002
Medical	BR	**.0117 ± .001**	.0123 ± .001	.0150 ± .003	.0179 ± .003	.0212 ± .001	.0150 ± .003
	LP	.0164 ± .005	.0186 ± .005	.0181 ± .005	.0208 ± .005	.0265 ± .001	.0185 ± .005
Scene	BR	.1374 ± .018	.1484 ± .012	.1577 ± .011	.1472 ± .012	.1292 ± .014	.1611 ± .010
	LP	.1636 ± .034	.1852 ± .016	.1710 ± .027	.1585 ± .024	**.1245 ± .026**	.1804 ± .019
Science	BR	**.0325 ± .001**	.0327 ± .001	.0341 ± .003	.0338 ± .001	.0353 ± .001	.0341 ± .003
	LP	.0483 ± .001	.0468 ± .001	.0437 ± .002	.0445 ± .001	.0461 ± .001	.0433 ± .001
Slashdot	BR	**.0439 ± .002**	.0452 ± .001	.0473 ± .003	.0483 ± .001	.0483 ± .002	.0476 ± .002
	LP	.0626 ± .004	.0620 ± .004	.0632 ± .004	.0706 ± .005	.0650 ± .003	.0623 ± .004
Social	RB	**.0216 ± .002**	.0224 ± .002	.0242 ± .005	.0242 ± .002	.0250 ± .001	.0245 ± .002
	LP	.0302 ± .003	.0286 ± .001	.0274 ± .002	.0274 ± .001	.0282 ± .001	.0273 ± .001
Yeast	BR	**.2068 ± .008**	.2078 ± .007	.2123 ± .009	.2133 ± .009	.2157 ± .007	.2116 ± .008
	LP	.2230 ± .011	.2273 ± .010	.2266 ± .011	.2268 ± .012	.2314 ± .011	.2245 ± .013

loss and *Subset 0/1 loss* metrics. Bold cells highlight the best performing algorithms for each data set.

Several conclusions may be drawn from these experiments:

- In the case of data sets in which a strong conditional dependence between labels is observed (all data sets except the Medical and Slashdot data sets [18, 21]), this result in different risk minimizers for both *Hamming loss* and *Subset 0/1 loss* metrics. One can observe for these data sets that feature selection methods treating each label independently (BRRF and BRMI here) are more appropriate for the *Hamming loss* compared to the ones that consider the interaction among labels for evaluating feature importance (RFLP, RFPCT, PMU and PPT-MI). More specifically, we observe that BRRF, used in tandem with BR as a multi-label base classifier, scores 12 wins and performs significantly better than BRMI. On the other hand, as far as the *Subset 0/1 loss* is concerned, the results suggest that it is more

TABLE 7.3: Subset 0/1 loss of all feature selection approaches and all data sets using BR and LP as base multi-label learning algorithm. Bold cells highlight the best performing algorithms for each data set.

Data set	ML Base learner	BRRF	BRMI	RFLP	RFPCT	PMU	PPT-MI
Arts	BR	.8471±.016	.8449±.035	.8802±.048	.8738±.031	.9259±.023	.8965±.041
	LP	.7774±.020	.7574±.027	**.7321±.036**	.7405±.032	.7788±.014	.7531±.029
Business	BR	.4584±.004	.4639±.002	.4586±.005	.4639±.003	.4651±.001	.4726±.072
	LP	.4578±.001	.4599±.001	**.4468±.005**	.4575±.003	.4570±.002	.4482±.005
Education	BR	.8649±.023	.8779±.024	.9246±.024	.9219±.012	.9281±.034	.9074±.032
	LP	.7595±.011	.7309±.024	.7193±.019	.7318±.006	.7146±.017	**.7097±.016**
Emotions	BR	.7950±.040	.8196±.043	.8135±.050	.8085±.038	.8644±.067	.8160±.060
	LP	.6935±.042	.6973±.038	**.6837±.018**	.6910±.013	.7361±.050	.7007±.038
Enron	BR	.9164±.025	.9455±.051	.9416±.046	.9518±.023	.9681±.027	.9858±.009
	LP	.8534±.014	.8836±.007	**.8447±.009**	.8504±.007	**.8444±.016**	**.8458±.005**
Entertainment	BR	.7394±.028	.7517±.039	.8107±.058	.8134±.066	.9412±.017	.8241±.075
	LP	.7236±.030	.6919±.036	**.6714±.044**	.6869±.039	.7422±.015	.6823±.040
Health	BR	.6477±.051	.6078±.026	.6771±.056	.6381±.024	.7081±.026	.6278±.018
	LP	.6770±.007	.6697±.017	.6191±.031	**.6050±.019**	.6312±.016	.6100±.018
Medical	BR	**.3931±.009**	**.4046±.012**	.5017±.100	.6014±.117	.7489±.028	.5038±.116
	LP	**.4067±.083**	.4599±.083	.4457±.083	.5204±.095	.5926±.030	.4574±.082
Scene	BR	.6811±.103	.7686±.061	.8073±.086	.7704±.065	.6198±.094	.8490±.057
	LP	.5134±.094	.5690±.046	.5230±.077	.5013±.069	**.4026±.073**	.5594±.054
Science	BR	.8739±.024	.8735±.023	.9194±.027	.9266±.024	.9843±.012	.9213±.027
	LP	.8063±.003	.7824±.011	.7354±.023	.7459±.016	.7701±.010	**.7274±.019**
Slashdot	BR	.7160±.038	.7429±.060	.7684±.050	.8707±.035	.8180±.046	.7664±.049
	LP	.6700±.031	**.6624±.034**	.6712±.032	.7324±.040	.6858±.022	.6641±.033
Social	BR	.5811±.060	.6101±.059	.6428±.098	.6526±.076	.6761±.056	.6375±.092
	LP	.5463±.050	.5142±.013	.4940±.032	.4923±.020	.5086±.013	**.4903±.017**
Yeast	BR	.8785±.035	.8915±.040	.9107±.045	.8972±.050	.9390±.030	.9002±.051
	LP	.7943±.024	.7959±.024	**.7866±.030**	.8001±.031	.8161±.023	.7962±.029

effective to use feature selection methods built considering the correlation among labels with LP as a multi-label base classifier, rather than ignoring this correlation within the feature selection process. In such case, the results show a relative superiority of RFLP which scores 6 wins, followed by PPT-MI (4 wins), then PMU (2 wins) and RFPCT (1 win). These results corroborate the previous finding in [18] for multi-label classification and extend them to the multi-label feature selection task.

- In the case of data sets (Medical and Slashdot) for which the labels are conditionally independent (see [18, 21] for more details about these data sets and their directed acyclic graphs (DAG)), it seems that both risk minimizers for *Hamming loss* and *Subset 0/1 loss* coincide. The best feature selection algorithms perform equally good for both losses. Here, BRRF and BRMI seem to have equivalent performances and perform significantly better than the remaining feature selection methods in terms of both *Hamming loss* and *Subset 0/1 loss*.

- Like in RFLP, RFPCT is also expected to take into account the interaction among labels for evaluating feature importance. However, RFPCT is still not well understood from a theoretical point of view. For example, it is not clear what loss function it intends to minimize compared to RFLP for which it is rather clear that it tries to minimize the *Subset 0/1 loss* metric [18]. The superiority of RFLP compared to RFPCT in the feature selection process could be further motivated by the following reasons. With RFPCT, the classification error does not vary significantly when the values of a specific feature are randomly permuted. Indeed, we noticed that the label errors often compensate each other. This is why the classification error vary moderately after shuffling a variable. This issue worsen as the number of labels is increased. To confirm this observation from an experimental point of view, we analyzed the average gap between classification error before and after the variable shuffling in Equation 7.1. We observed error variations of the magnitude of 10^{-7} on the data sets with a large number of labels (*e.g.* Enron, Medical).

- More generally, these experiments confirm the ability of Random Forest, that showed promising results for multi-label classification in [139], to rank the relevant features accurately in a multi-label context.

7.2.3 Robustness analysis of feature selection

In this section we report on the experiments performed to evaluate the robustness of aforementioned feature selection methods. The robustness of feature selection techniques can be defined as the variation in feature selection results due to small changes in the data set. When applying feature selection for knowledge discovery, not only model performance but also robustness of the feature selection process is important, as domain experts would prefer a stable feature selection algorithm over an unstable one when only small changes are made to the data set [140]. Robust feature selection techniques would allow domain experts to have more confidence in the selected features, especially if subsequent analyses or validations of selected feature subsets are costly.

To assess the robustness of the compared multi-label feature selection techniques, we focus here on comparing feature rankings using the conventional consistency index I_C in [141] for the top 5% features of the rankings obtained over the 15 iterations. The Consistency Index for two feature subsets S_i and S_j, such that $\left|S_i\right| = \left|S_j\right|$ is given by,

$$I_C(S_i, S_j) = \frac{rM - k^2}{k(M - k)} \tag{7.2}$$

The overall stability of a feature selection algorithm for a set of sequences of features $\mathcal{A} = \{S_1, S_2, \ldots, S_K\}$ ($K = 15$ in our case) is defined as the average over all pairwise consistency indices:

$$Robustness = \frac{2\sum_{i=1}^{K-1}\sum_{j=i+1}^{K} I_C(S_i, S_j)}{K(K-1)} \tag{7.3}$$

where M is the number of features in the data set, $k = |A| = |B|$ and r is the cardinality of the intersection of subsets A and B. The more similar the outputs, the higher the stability measure.

Table 7.4 summarizes the results of the robustness analysis across the different data sets. The conclusions we can draw upon looking at this table follows:

1. Overall, BRRF exhibits more robust results than the other algorithms. Indeed, BRRF clearly benefits from averaging of feature importances over the different forests (one forest per label), hence the gain in robustness of the feature ranking. BRRF is followed by RFLP. This demonstrates again the effectiveness of ensemble methods to improve the robustness of the feature selection [140].

2. RFPCT is however the less stable algorithm. This is especially due to our aforementioned observation, namely that when estimating feature importance with RFPCT the classification error vary moderately after shuffling a variable, resulting in very small variations across the feature importances. This leads to a degradation in the robustness because the top performing features vary a lot with respect to the data subsamples. The situation worsen as the number of labels is increased. As may be observed, the robustness of RFPCT on *Enron* decreased dramatically. The large variance among the top selected features is the main caveat of RFPCT.

3. PPT-MI on the other hand proves to be more stable compared to PMU and BRMI.

7.3 Chapter summary

This Chapter presented and experimentally evaluated three wrapper multi-label feature selection methods, which use the Random Forest paradigm: BRRF, RFLP and RFPCT. These extensions differ in the way they consider label dependence within the feature selection process. The performance of the methods were compared against recently proposed approaches using 13 benchmark multi-label data sets emerging from different domains. The result of this evaluation is two-fold: 1) Random Forest handles accurately the feature selection process in a multi-label context and is able to improve the efficiency as well as the robustness of feature selection techniques; 2) We also demonstrates how the optimized loss function in the multi-label classifier influences the relevance of a multi-label feature selection process, thereby contributing to a better understanding of the internal meaning of selected features. According to this analysis, BRRF appears more suitable

TABLE 7.4: Robustness of the different multi-label feature selection methods across the different data sets using the consistency index on the subset of 5% best features.

Data set	BRRF	BRMI	RFLP	RFPCT	PMU	PPT-MI
Arts	0.934	0.694	0.848	0.9	0.742	0.779
Business	0.826	0.694	0.783	0.269	0.696	0.59
Education	0.88	0.681	0.809	0.34	0.675	0.843
Emotions	0.765	0.62	0.718	0.612	0.406	0.575
Enron	0.763	0.635	0.722	0.174	0.544	0.496
Entertainment	0.962	0.708	0.917	0.68	0.719	0.945
Health	0.852	0.739	0.82	0.379	0.743	0.716
Medical	0.9	0.769	0.82	0.546	0.62	0.874
Scene	0.856	0.856	0.574	0.572	0.577	0.637
Science	0.862	0.647	0.776	0.375	0.615	0.696
Slashdot	0.835	0.688	0.768	0.468	0.661	0.874
Social	0.908	0.683	0.81	0.508	0.688	0.891
Yeast	0.906	0.713	0.792	0.744	0.727	0.622
Average	0.875	0.723	0.788	0.526	0.652	0.753

for label-wise metrics (like *Hamming loss*), while RFLP is more appropriate for instance-wise metrics such as *Subset 0/1 loss*, in the case of data sets in which a strong conditional dependence between labels is observed. RFPCT on the other hand is still not well understood from a theoretical point of view and it is rather unclear what this approach actually tends to optimize.

In the next Chapter we consider the problem of using a large amount of unlabeled data to improve the efficiency of feature selection in high dimensional multi-label data sets, when only a small set of labeled examples is available. The way internal estimates are used to measure variable importance in the Random Forest paradigm and discussed in this Chapter have been influential in our thinking. We extended our previously proposed k-labelsets based ensemble approach CkMLC [81] (*c.f.* Chapter 4) to deal with multi-label feature selection in a semi-supervised context by using both labeled and unlabeled data. Consequently, we propose a new semi-supervised multi-label feature importance evaluation method (SSkC for short), that combines ideas from co-training and random k-labelsets ensemble learning with a new permutation-based out-of-bag feature importance measure.

Chapter 8

Semi-Supervised k-labelsets ensemble framework

Similarly to other machine learning tasks, multi-label learning also experiences the curse of dimensionality, which may cause problems when learning from high-dimensional data. The identification of relevant subsets of random variables (*i.e.* feature selection), among thousands of potentially irrelevant and redundant variables, is a very important issue to overcome this problem. In this regard, feature selection algorithms use information from labeled data to find the relevant subsets of variables, *i.e.*, those that conjunctively prove useful to construct an efficient classifier from data. They enable the classification model to achieve good or even better solutions with a restricted subset of features [96]. As discussed in Chapter 6, Multi-label feature selection has been widely studied and have encountered some success in many applications during the past few years [104, 111, 114]. In multi-label learning, most feature selection tasks have been addressed by extending the techniques available for single-label classification using either the bridge provided by the multi-label transformations or new adaptation approaches.

These methods have been designed to work with a sufficient amount of labeled training data. However, in many real-world applications, the amount of labeled data is very limited, in the sense that it is time-consuming or extremely expensive to obtain. In such situation, there are mainly two challenges. First, in the presence of few amount of labeled data, it becomes difficult to build an accurate multi-label model. Meanwhile a large amount of unlabeled data may be relatively easy to collect, but there has been few ways to use them. Semi-supervised multi-label learning addresses this problem by using unlabeled data together with multi-labeled data in the training process, to enhance the performance of the learned classifiers. The second challenge is that the labels in multi-label learning are typically interdependent and correlated, which poses more difficulties to identify or remove redundant and irrelevant variables from the feature set, especially in high-dimensional data. To overcome this problem, feature selection methods need

to explicitly model the label interactions in evaluating the quality of features, which is crucial for better performance.

In this Chapter, based on the above motivation, we aim to solve both challenges in one shot. We present a new ensemble approach for semi-supervised multi-label feature selection that use both dependencies between labels and the unlabeled data together to enhance the multi-label learning performance. It ranks features through a multi-label ensemble framework, in which a feature's relevance is evaluated by its predictive performance using both labeled and unlabeled data. The proposed approach, termed as *Semi-Supervised k-labelsets Committee* (SSkC) [142] extends our k-labelsets based ensemble model CkMLC [81] (*c.f.* Chapter 4) to handle semi-supervised multi-label feature selection. It combines both data resampling (bagging) and random projections of the label space (random k-labelsets) strategies for generating a committee of multi-label models in a co-training style algorithm. The key ideas behind this approach are to i) promote and maintain diversity in the multi-label base-classifiers committee, ii) define a new cost oriented metric to estimate the labeling confidence of unlabeled examples, and iii) use a new multi-label permutation-based out-of-bag feature importance measure which operates over both labeled an unlabeled instances in a semi-supervised way.

In the rest of this Chapter, we first introduce the SSkC framework for variable importance estimation. Then, we present our experiments using relevant multi-label benchmarks data sets to compare SSkC to a recent state-of-the-art supervised and semi-supervised multi-label feature selection algorithms over different multi-label metrics.

8.1 The proposed framework

One of the most attractive semi-supervised ensemble models is the Co-training algorithm [143]. In Co-training two base-classifiers are initially trained using two redundant and independent sets of features. Then, in further iterations, each base-classifier classifies the unlabelled examples, adds the examples about which it is most confident in the training set. The aim is that the most confident examples with respect to one classifier can be informative with respect to the other. As an improvement of the Co-training algorithm, Hady and Schwenker proposed the Co-training By Committee (CoBC) learning approach [144]. In this model, an ensemble of diverse base-classifiers is used instead of redundant and independent views. The committee of diverse accurate classifiers is initially constructed by using a successful ensemble learning algorithms: Bagging or random subspace method. At each iteration and for each classifier, a subset of unlabelled examples is drawn randomly from the whole unlabelled data set and classified using the concomitant ensemble. The most confident examples to label are then determined and the committee members are retrained using their updated training sets.

On the other hand, as aforementioned before, semi-supervised multi-label feature selection have encountered some success during the past few years. However, no attention has been given to exploiting the power of ensemble methods with a view to identify and remove the irrelevant features in a semi-supervised multi-label setting. Such methods which combines multiple base learners to jointly accomplish one common task are shown to be very beneficial for enhancing the robustness and the generalization ability of single learners and overcoming the curse of dimensionality problem. In this section, we discuss in details our semi-supervised multi-label ensemble Learning Guided feature selection framework, named SSkC [142]. It combines ideas from *co-training*, *bagging*, and *random k-labelsets ensemble learning* with an extension of the RF permutation importance measure.

8.1.1 Committee construction

While considerable attention has been given on the problem of constructing an accurate and diverse ensemble committee for multi-label learning [15, 17] and to the best of our knowledge this is the first attempt that tries to explore this strategy in the semi-supervised multi-label learning.

Given a set of multi-labeled training examples L associated with a set of labels in $\mathcal{L} = \{\lambda^1, \lambda^2, ..., \lambda^q\}$ and a set of unlabeled training examples U, independently drawn from the same data distribution and described over the input space $F = \{f_1, \ldots, f_p\}$, our approach SSkC constructs a committee according to the following steps.

The implementation of our ensemble k-labelsets model is based on the top of T multi-label base-classifiers, where each classifier is trained on a small subset of k labels from $\mathcal{L}$ as in [15]. As discussed before, the most important condition for a successful ensemble learning method is to combine models which are different from each other. Thus, to maintain diversity between committee members, we have employed two strategies : data resampling (*bagging*) of labeled instance set L and random projections of the label space (random k-labelsets). A combination of these two main strategies for producing ensemble of classifiers leads to exploration of distinct views of inter-pattern relationships. To further maintain the diversity during the learning process in the semi-supervised setting, we also use the bagging strategy over the set of unlabelled instances U. The objective here is to keep the diversity over the augmented training set for the retrained multi-label classifiers once the most confident unlabeled data are incrementally added into the labeled data set.

The formal description of SSkC is given in Algorithm 5. First, as formulated in step A, the initial committee is constructed as follows: for each committee member h_t, a k-labelsets (PS_t) is formed with k labels randomly selected from $\mathcal{L}$. Then, L_t^{bag} and U_t^{bag} are selected with replacement, from L and U respectively. Each base-classifier h_t is learned by a ML-*BaseLearner* using its corresponding labeled training examples L_t^{bag} and its corresponding k-labelsets PS_t. The

ensemble model learned by our approach output a score vector and need a thresholding method in order to assign for each unlabeled instance a label set in $\mathcal{L}$. In step 11, our algorithm is used in conjunction with our previously proposed *Forward Multi-label Thresholds Calibration* method (*c.f.* Algorithm 3 in Chapter 4) that optimizes a multi-label performance measure of interest (ML-$loss$). Step 12 uses a new permutation-based out-of-bag feature relevance measure which operates over the out-of-bag instances in order to give a first accurate rank of feature importances per label.

The block B identifies the concomitant ensemble of each base-classifier h_t. Denoted by $c\text{-}H_t$, the concomitant ensemble of h_t is formed by all the classifier members of the committee H sharing at least one label $\lambda \in \mathcal{L}$ with h_t, *i.e.*, $c\text{-}H_t = \{h_i \in H | \exists \lambda \in \{PS_i \cap PS_t\}$ with $t \neq i\}$.

Finally, according to the steps in C, each committee member h_t is trained in a co-training style by asking its concomitant ensemble $c\text{-}H_t$ to label samples from U_t^{bag} for it. In order to avoid that the concomitant ensemble gives a biased labels prediction, each concomitant member is asked to label only its out-of-bag instances, *i.e.*, instances that do not appear in its bag and are never used to learn this classifier member. Thereby, the number of labeled examples for each base-classifier increases by including the most confident new labeled examples for the k-labelsets PS_t. To describe how the most confident examples are selected a formal description of the $SelectConfidantExamples$ function is given in Algorithm 6. Next, the newly labeled samples Π_t^* for h_t are removed from U_t^{bag}, and incrementally added into its set of labeled instances $\hat{L}_t$. Afterwards, the multi-label base-classifier h_t is retrained over the augmented set $L_t^{bag} \cup \hat{L}_t$.

Our ML-$BaseLearner$ can use any learning algorithm for training each classifier h_t ($t \in \{1, \ldots, T\}$). It is worth noting that our approach produces relevance scores of features in F. In this incremental retraining process, instead of considering equally the features when training a given committee member h_t, we suggest to randomly select the features according to their relevances in predicting accurately its corresponding k-labelsets PS_t. Our ensemble approach relies on this step to simultaneously encourage diversity and individual accuracy in the committee. The goal of this selection scheme is to consider the feature subspaces which are as relevant as possible to the k-labelsets PS_t, especially for large p. Using probability of selection proportional to relevance scores ensures that informative features are selected and will lead to promote the accuracy of the committee members. On the other hand, since that different base-classifier focus on different k-labelsets having their specific relevant features, the use of feature importance in our approach will maintain the randomness in our committee construction and does not hurt the diversity of classifiers.

Once all the base-classifiers are updated, the label decision thresholds are re-calibrated to meet the objective multi-label performance measure of interest (step 22 using our *Forward Multi-label Thresholds Calibration* method) and the feature importances are re-evaluated (step 23) using

both labeled and unlabelled instances. Finally, the co-training steps are repeated until a maximal number of iteration is reached.

8.1.2 Confidence measure

One of the most important aspects in a co-training style approach is how to estimate the label confidence of unlabeled instances which gives their probabilities of being selected. Indeed, an inaccurate confidence measure leads to adding noisy instances to the labeled training set. Algorithm 6 gives a formal description of how the most confidant instances are selected in our framework. More specifically, to efficiently estimate the confidence of an unlabeled instance $\mathbf{x}_u$ for a base-classifier h_t ($\mathbf{x}_u \in U_t^{bag}$), each classifier member h^j in the concomitant ensemble $c\text{-}H_t$ which did not use $\mathbf{x}_u$ in its training process ($\mathbf{x}_u \in U_j^{bag}$) is asked to label it and to generate an estimation of the probability $\hat{P}(y^i = 1|\mathbf{x}_u)$ of having label λ^i (for each $\lambda^i \in PS_t$) given $\mathbf{x}_u$. Thus, the probability $S^i(\mathbf{x}_u)$ for $\mathbf{x}_u$ of having the label λ^i is estimated through averaging all base-classifiers scores in $c\text{-}H_t$. Nevertheless, label distribution in multi-label classification is highly imbalanced. An accurate decision threshold could be different from the traditional single threshold 0.5 and may change also from one label to another. In the previous Chapters, we have shown that threshold calibration can improve dramatically the multi-label performances, especially when the calibration is in line with an objective multi-label loss function of interest. Therefore, it is wise to consider the decision threshold for each label when selecting the final predicted labelset of $\mathbf{x}_u$ ($y_{\mathbf{x}_u}$) and in estimating its confidence. This will firstly help to tackle the imbalance label distribution problem and secondly to keep the confidence measure consistent with an optimized performance metric. The confidence measure of an unlabeled instance x_u given a label λ_i with a threshold t_i can be defined as follows:

$$Conf^i(S^i(\mathbf{x}_u), \tau^i) = \frac{|S^i(\mathbf{x}_u) - \tau^i|}{\delta(S^i(\mathbf{x}_u), \tau^i)}$$

where $S^i(\mathbf{x}_u)$ is the estimation of the probability of having the label λ^i for $\mathbf{x}_u$ and

$$\delta(z, \tau^i) = \begin{cases} \tau^i & \text{if } z \leq \tau^i \\ 1 - \tau^i & \text{if } z > \tau^i \end{cases}$$

Our confidence measure is based on the margin between the decision threshold and the estimated label score $S^i(x_u)$. Consequently, the confidence of a committee on predicting a labelset related to a vector of thresholds $\tau = (\tau^1, \tau^2, \cdots, \tau^q)$ is given by:

$$Confidence(S(\mathbf{x}_u), \tau) = min(Conf^1, Conf^2, \cdots, Conf^q)$$

Algorithm 5 *Semi-supervised k-labelset model*

Require:

Training Multi-label samples (L); Unlabelled training examples (U); Maximum number of iterations ($maxiter$); Multi-label base-learner (ML-$BaseLearner$); k-labelsets size (k); Ensemble size (T); Number of instances to label (n); Multi-label loss function (ML-$loss$); Set of feature space descriptors ($F = \{f_1, \ldots, f_p\}$)

$H \leftarrow \emptyset$

$Fimp(i, j) = \frac{1}{p}$ (for $i = \{1, \ldots, p\}$ and $j = \{1, \ldots, q\}$)

A- Initial Committee construction

for $t = 1 : T$ **do**

  $PS_t =\leftarrow$ randomly draw k labels from $\mathcal{L}$

  $L_t^{bag} \leftarrow$ bootstrap sample from L

  $U_t^{bag} \leftarrow$ bootstrap sample from U

  $L_t^{oob} \leftarrow L \backslash L_t^{bag}$; $U_t^{oob} \leftarrow U \backslash U_t^{bag}$

  $h_t \leftarrow$ ML-$BaseLearner(L_t^{bag}, PS_t, Fimp)$

  $H \leftarrow H \cup h_t$

end for

$\tau \leftarrow ThresholdCalibration\ (H, L^{oob}, \text{ML-}loss)$

$Fimp \leftarrow MeasureFeatureImportance(H, L^{oob}, U^{oob})$

B- Co-committee identification

for $t = 1 : T$ **do**

  $c\text{-}H_t \leftarrow \{h_i \in H | \{PS_i \cap PS_t\} \neq \emptyset \text{ with } t \neq i\}$

end for

C- Committee refinement

for $iter = 1 : maxiter$ **do**

  for $t = 1 : T$ **do**

    $\prod_t^* \leftarrow SelectConfidantExamples(c\text{-}H_t, U_t^{bag}, \tau, n, \text{ML-}loss)$

    $U_t^{bag} = U_t^{bag} \backslash \prod_t^*$; $L_t^{bag} \leftarrow L_t^{bag} \cup \prod_t^*$

    $h_t \leftarrow$ML-$BaseLearner(L_t^{bag}, PS_t, Fimp)$

  end for

  $\tau \leftarrow ThresholdCalibration\ (H, L^{oob}, \text{ML-}loss)$

  $Fimp \leftarrow MeasureFeatureImportance(H, L^{oob}, U^{oob})$

end for

return H and $Fimp$

Once the multi-label confidence measure is computed for all unlabeled examples in U_t^{bag}, the n top-ranked labeled instances along with their corresponding labels are selected as a candidate instances to expand the set of h_t's labeled samples.

Algorithm 6 *Select Confident Examples*

Require:

Concomitant ensemble (c-H_t); Unlabeled data set (U_t^{bag}); Multi-label decision threshold (τ)
Number of most confident instances to select (n); Multi-label loss function (ML-$loss$);

1: **for** each $x_u \in U_t^{bag}$ **do**
2: $S(\mathbf{x}_u) \leftarrow$ predict $\mathbf{x}_u$ using its out-of-bag c-H_t
3: $y_{\mathbf{x}_u} \leftarrow$ threshold $S(\mathbf{x}_u)$ using τ
4: $Conf(\mathbf{x}_u) \leftarrow$ Confidence($S(\mathbf{x}_u), \tau$)
5: **end for**
6: $\Pi_t \leftarrow$ select the top n ranked instances in U_t^{bag} along with their corresponding labels
7: $\Pi_t^* \leftarrow$*NoiseElimination*(Π_t, ML-$loss$)
8: **return** Π_t^*

One of the most important problems of semi-supervised learning resides in the noise brought by unlabeled data. Explicitly, false-labelled instances accepted in the training set, serve as correct instances and hurt the classification quality. Compared to traditional single-label learning, the problem is more challenging in multi-label context since it affects a set of labels. In order to reduce this effect, it is important to efficiently remove the noisy instances. In our approach, the newly labeled instances go throughout a noise elimination procedure which take advantage from the out-of-bag labeled data set L^{oob} of each committee member h_t. The basic assumption is that a correctly labeled instances should not hurt the classification performance of h_t regarding the multi-label performance measure of interest (ML-$loss$). Here, this can be achieved by the Backward-Froward search strategy. The detailed description of our multi-label noise elimination procedure is given in Algorithm 7.

In detail, the search strategy starts by evaluating an unbiased performance of the committee member h_t over the L^{oob} data when trained with the complete set of candidates Π_t and compare it to the original performance (without adding the newly labeled instances). If the model performance are not improved, then the search strategy tries either to remove N_{out} instances from Π_t to be added to a set of potential noisy instances set $\overline{\Pi_t}$, or to reintroduce N_{in} instances from $\overline{\Pi_t}$ to the set of candidates instances Π_t where $N_{out} > N_{in}$. This process is repeated until the model's performance improves or remains steady. Due to the bootstrapping strategy on unlabeled data in our framework, it is notable that some instances in U_t^{bag} may occur multiple times. In order to guarantee the consistency of the learning process and an accurate labeling for unlabeled data, we consider all the occurrences of the same instance in noise elimination step as a single example in each iteration.

Algorithm 7 *Backward-Forward noise elimination*

Require:

Committee member (h_t); Out-of-bag labeled samples (L_t^{oob}) Set of new labeled instances (Π_t); Multi-label loos function (ML-$loss$); Search rate (r)

$\hat{y} \leftarrow$ predict L_t^{oob} using h_t
$e \leftarrow$ ML-$loss(\hat{y}, L_t^{oob})$
$h^* \leftarrow$ update h_t using L_t^{bag}; $\hat{U}_t$ and Π_t
$\hat{y}^* \leftarrow$ predict L_t^{oob} using h^*
$e^* \leftarrow$ ML-$loss(\hat{y}^*, L_t^{oob})$
$flag \leftarrow 0$; $\Pi_{out} \leftarrow \emptyset$
while $e < e^*$ **do**
  if $flag < r$ **then**
    $flag \leftarrow flag + 1$
    $\pi \leftarrow$ randomly select a sample from Π_t
    $\Pi_{out} \leftarrow \Pi_{out} \cup \pi$
    $\Pi_t \leftarrow \Pi_t \backslash \pi$
    $h^* \leftarrow$ update h_t using L_t^{bag}; $\hat{U}_t$ and Π_t
    $\hat{y}^* \leftarrow$ predict L_t^{oob} using h^*
    $e^* \leftarrow$ ML-$loss(\hat{y}^*, L_t^{oob})$
  else
    $flag \leftarrow 0$
    $\pi \leftarrow$ randomly select a sample from Π_{out}
    $\Pi_t \leftarrow \Pi_t \cup \pi$
    $\Pi_{out} \leftarrow \Pi_{out} \backslash \pi$
    $h^* \leftarrow$ update h_t using L_t^{bag}; $\hat{U}_t$ and Π_t
    $\hat{y}^* \leftarrow$ predict L_t^{oob} using h^*
    $e^* \leftarrow$ ML-$loss(\hat{y}^*, L_t^{oob})$
  end if
end while
$\Pi^* \leftarrow \Pi_t$
return Π_t^*

8.1.3 Out of Bag multi-label feature relevance measure

The key for designing an effective semi-supervised feature selection algorithm is to develop a framework under which the feature importance is measured using both labeled and unlabeled samples in a natural way. In our approach, the random projections of the label space method is combined to bootstrapping. Actually, in each bootstrapped labeled and unlabeled set, almost 33% are left oob, i.e., they are not used for the construction of the corresponding model. We refer to them as L_t^{oob} and U_t^{oob}. Thus, these patterns can be used to estimate an unbiased feature importance. Our proposed feature selection measure is based on the assumption that a feature f is relevant for the classification of a label λ_i if small variation over f leads to a shifted predictions over the label λ_i. Thus, the importance of the feature f can be measured by the number of correctly predicted instances that changes classification when the values of feature f are randomly permuted. Clearly, for the labeled examples, a label λ_i is well predicted, if the label assigned by h_t corresponds to the real label. Its label confidence is set to 1. For unlabelled examples, the right label is unknown. The idea in this work is to assume that an unlabelled example $\mathbf{x}_u$ is "well labeled" by h_t if the label given by h_t is the label given by the ensemble committee H. In that case, the label confidence will be set to $Conf^i(S^i(\mathbf{x}_u), \tau^i)$ as in the previous section.

The feature importance procedure works in two steps: the first step computes the feature importance within the set of labeled instances L^{oob} whereas the second step focuses on unlabeled instances U^{oob}. Algorithm 8 summarizes the procedure. To estimate the importance of a feature f, the values of the feature f are randomly permuted over the *oob* samples in L and U. We refer to these subsets by $\mathcal{R}L_t^{oob}$ and $\mathcal{R}U_t^{oob}$. Over the both steps, each committee member h_t is used to predict the k-labelsets of the new switched out-of-bag instances. Then, for each label λ_i in PS_t the sum of all the miss-labelled example's confidence is computed. The latter value is summed for each label λ_i over the T classifiers in the committee and the resulting value is taken as the global importance of the feature f. The procedure is repeated for every feature $f \in \{f_1, \ldots, f_p\}$.

8.1.4 Why should our approach work

The proposed SSkC framework enjoys several advantages.

First, SSkC takes advantage from the unlabeled instances to generate a committee of diverse classifiers. This characteristic improves the generation ability of the SSkC model compared to supervised k-labelsets based approaches such as RAkEL[15] and CkMLC [81] or TREMLEC [79], especially when the available labeled training set is small within a large feature space. Indeed, a supervised k-labelsets based ensemble relies on the available training data for encouraging diversity and enhancing base-classifier accuracy. So, if the size of the training set is as small as for semi-supervised settings, performing an efficient LP base-classifier will be a hard task.

Algorithm 8 *Feature Importance Measure*

Require:

Semi-supervised k-labelsets model H; Out-of-bag labelled samples (L_t^{oob}); Out-of-bag unlabelled samples (U_t^{oob}); Multi-label decision threshold (τ); Set of feature space descriptors ($F = \{f_1, \ldots, f_p\}$)

Return:

Label feature importance $Fimp$

Feature importance in L

$Imp_L \leftarrow 0$
for $f \in F$ **do**
  for $h_t \in H$ **do**
    $\hat{y} \leftarrow$ predict L_t^{oob} with h_t
    $\mathcal{R}L_t^{oob} \leftarrow$ randomly permute f in L_t^{oob}
    $\hat{y}_{PS_t} \leftarrow$ predict $\mathcal{R}L_t^{oob}$ with h_t
    Increase $Imp_L(f, PS_t)$ by the number of mismatches between $\hat{y}$ and $\hat{y}_{PS_t}$ over each label
  end for
end for

Feature importance in U

$Imp_U \leftarrow 0$
for $f \in F$ **do**
  for $h_t \in H$ **do**
    $S \leftarrow$ predict U_t^{oob} with the out-of-bag c-H_t
    $\hat{y} \leftarrow$ threshold S using τ
    $Conf(U_t^{oob}) \leftarrow$ Confidence(S, τ)
    $\mathcal{R}U_t^{oob} \leftarrow$ randomly permute f in U_t^{oob}
    $S_{PS_t} \leftarrow$ predict $\mathcal{R}L_t^{oob}$ with h_t
    $\hat{y}_{PS_t} \leftarrow$ threshold S_{PS_t} using τ
    Increase $Imp_U(f, PS_t)$ by the label confidence $Conf$ of mismatches between $\hat{y}$ and $\hat{y}_{PS_t}$ over each label
  end for
end for

Global Feature importance

$Fimp \leftarrow Imp_L + Imp_U$
return $Fimp$

Second, SSkC maintains the diversity within the ensemble committee throughout the co-training process. This diversity is sustained both by the bagging over U and also by allowing to each base-classifier to focus only on the most relevant features for its k-labelsets which tackles the curse of dimensionality problem in the input space.

Third, the objective loss function optimized by the SSkC is well defined and consistent through every step of the ensemble construction (*i.e.* the aggregation of the committee prediction, the confidence on unlabelled data and the feature importance evaluation). In addition, this cost function alignment remains unbiased through the use of out-of-bag and also allows to take into account the imbalance label representation in multi-label data.

8.2 Performances analysis

This section shows empirical results on benchmark multi-label data sets and compare SSkC against state-of-the-art semi-supervised and supervised multi-label feature selection algorithms. SSkC is compared with three other feature selection methods : (1) the greedy forward feature selection algorithm PPT-MI which is a filter multi-label feature selection method based on multidimensional Mutual Information [109], (2) the Convex Semi-supervised multi-label Feature Selection (CSFS) [123], and (3) the recent *soft-constrained Laplacian score* multi-label feature selection method (S-CLS). Seven benchmark multi-label data sets, obtained from the *Mulan's repository* [87], were used to assess performance of SSkC. The selected data sets cover different application domains: Biology, semantic scene analysis and music emotions. Table 8.1 summarizes basic statistics of the data sets: the number of examples **N**; the number of features **M**, the number of labels **q**; the Label Cardinality **Card**= $\frac{1}{N}\sum_{i=1}^{N}|Y_i|$, which is the average number of labels associated with each example; the Label Density **LD**= $\frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i|}{Q}$, which is the normalized **Card**.

TABLE 8.1: Description of the multi-label data sets used in the experiments.

Data	**Domain**	**N**	**M**	**q**	***Card***	**LD**
Business	Yahoo-Text	5000	438	30	1.588	0.053
Education	Yahoo-Text	5000	550	33	1.460	0.044
Emotions	Music	593	72	6	1.869	0.311
Entertainment	Yahoo-Text	5000	640	21	1.420	0.068
Health	Yahoo-Text	5000	612	32	1.662	0.052
Scene	Image	2407	294	6	1.074	0.179
Yeast	Biology	2417	103	14	4.237	0.303

8.2.1 Evaluation framework

To make fair comparisons, parameters for each algorithm were set as suggested in the literature for yielding the most satisfactory performances. For our SSkC approach, the size of k-labelsets k was set to 3 as in our gold standard ML ensemble approach RAkEL [15] and the *classregtree* Matlab implementation of decision tree is used for training the LP base-classifiers. The committee size T was computed using the following formula: $T = 10 \times ceil(\log(\alpha)/\log(1-1/k))$. This formula ensures that each label is drawn 10 times at a confidence level of $\alpha = 1\%$.

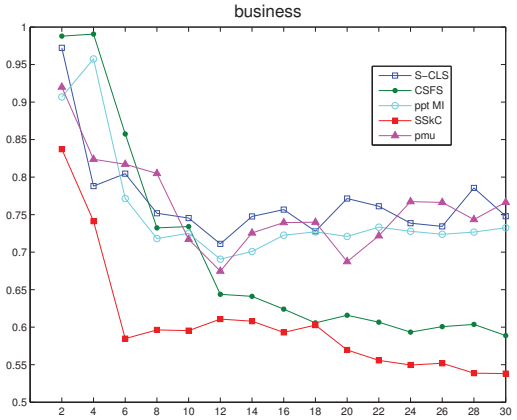
Regarding the number of iterations *maxiter* and the sample size n in our approach SSkC, they are both set to 10. For PPT-MI, the numeric data sets are discretized using the Equal width interval scheme, as suggested by the authors in [111]. The regularization parameter μ of the CSFS was tuned in the range of $\{10^{-6}, 10^{-4}, 10^{-2}, 10^{0}, 10^{2}, 10^{4}, 10^{6}\}$ so to report the best results as in [123]. In S-CLS the regularization parameter was set as suggested by the authors [124] and the k-neighborhood parameter is set to 10 for all data sets.

Moreover, the 2-fold cross validation is used to evaluate the performance of the compared methods. To get reliable statistics over the performance metrics, experiments were repeated 25 times. So, the results obtained were averaged over 50 runs. To simulate a semi-supervised context in each iteration, we randomly select 10% of instances from the training fold as labeled data, while the remaining training instances are used as unlabelled data.

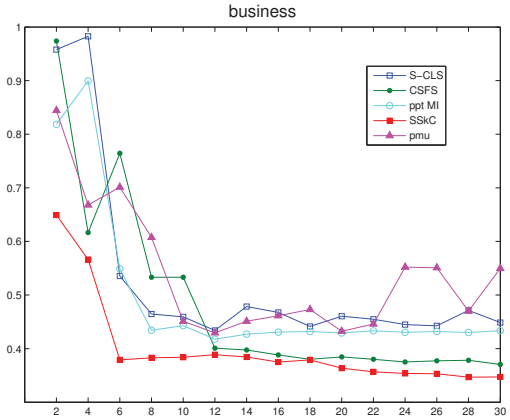
In order to assess the quality of a feature subset obtained with the aforementioned semi-supervised procedures, we train the semi-supervised algorithm TRAM [145] using the labeled data and the unlabelled data, and evaluate its performances on the test data according to six multi-label measures, *Subset 0/1 loss*,*Jaccard loss*,*Instance-F1 loss*,*Micro-F1 loss*, *Micro-F1 loss* and *Hamming loss*. The obtained measure is taken as the score for the feature subset. We preferred to assess the feature selection quality over a semi-supervised algorithm because it reflects the condition in which these variables are supposed to be used. Moreover, it is worth noting that our approach SSkC is performed six runs, each of them using one evaluation metric as an objective multi-label performance measure of interest (ML-$loss$) in the threshold calibration method.

8.2.2 Results

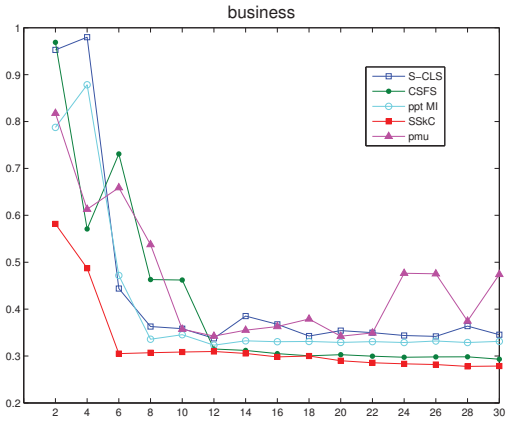
FIGURE 8.1: Performances metrics averaged over the 25x2 runs vs. different numbers of selected features on Business data set.



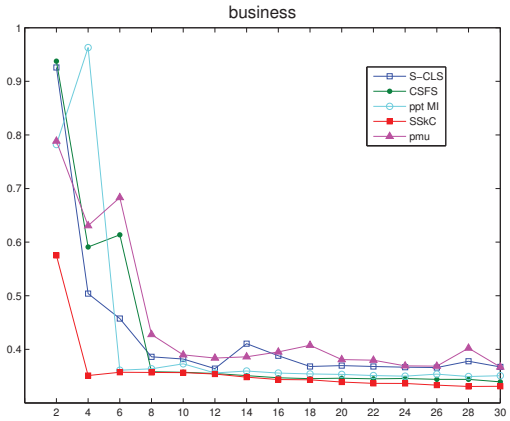
(a) Subset 0/1 loss



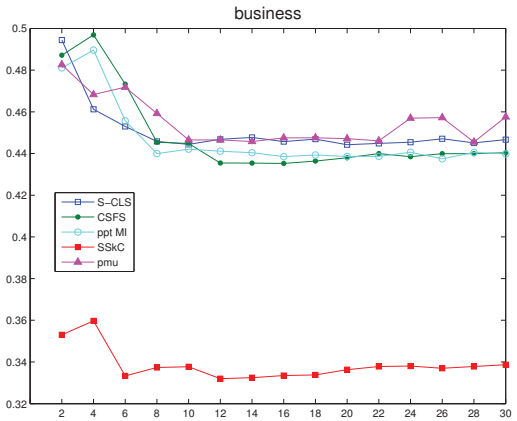
(b) Jaccard loss



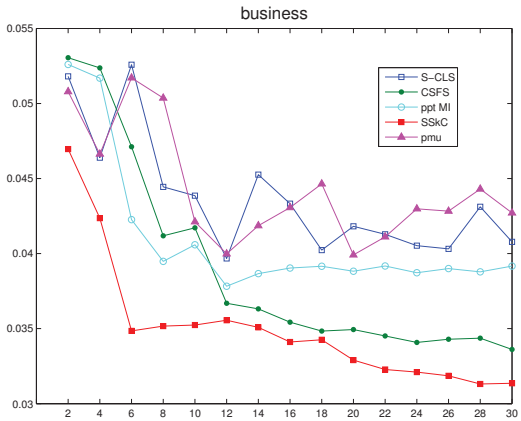
(c) Instance-F1 loss



(d) Micro-F1 loss

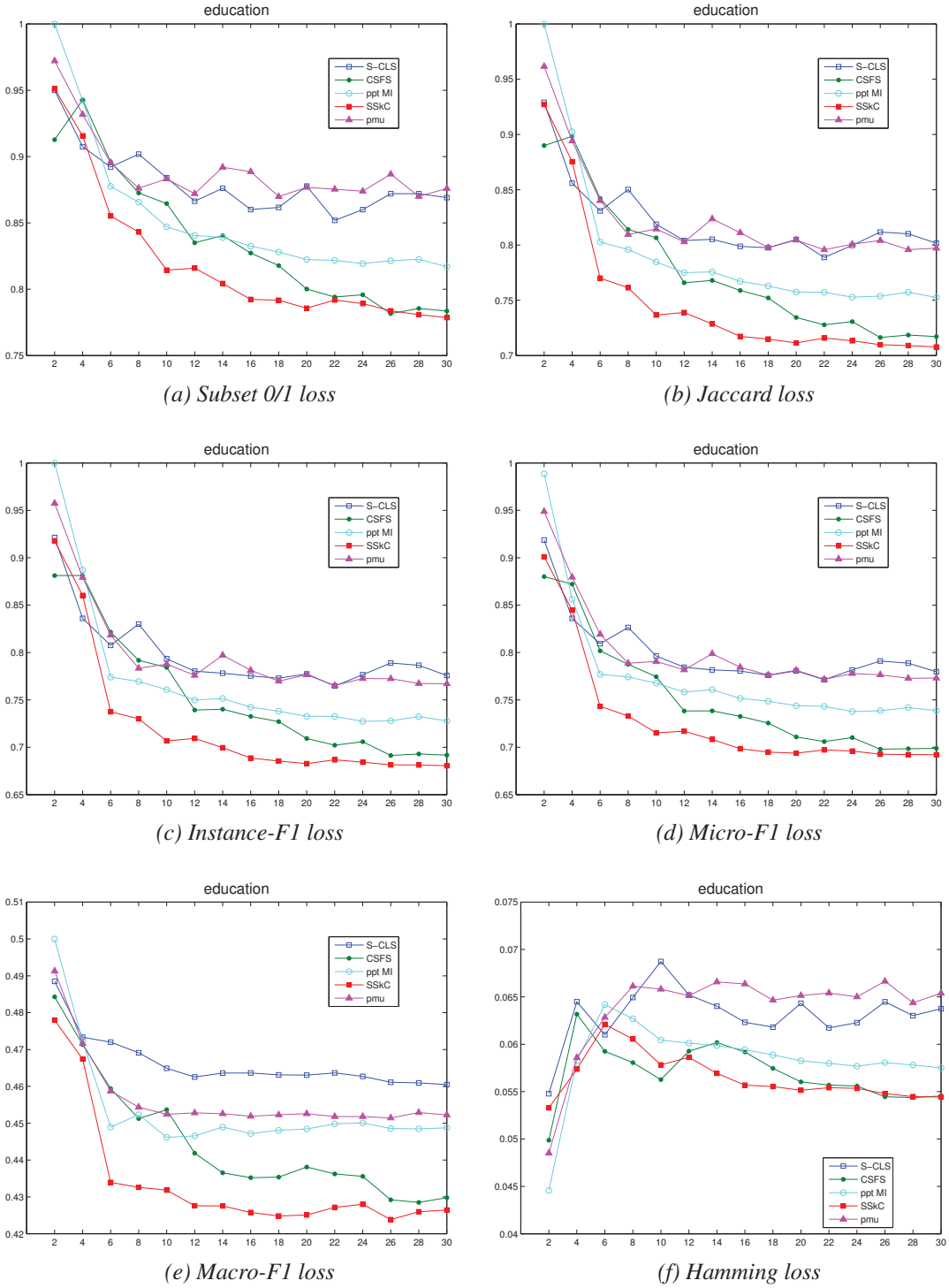


(e) Macro-F1 loss



(f) Hamming loss

FIGURE 8.2: Performances metrics averaged over the 25x2 runs vs. different numbers of selected features on Education data set.



(a) Subset 0/1 loss

(b) Jaccard loss

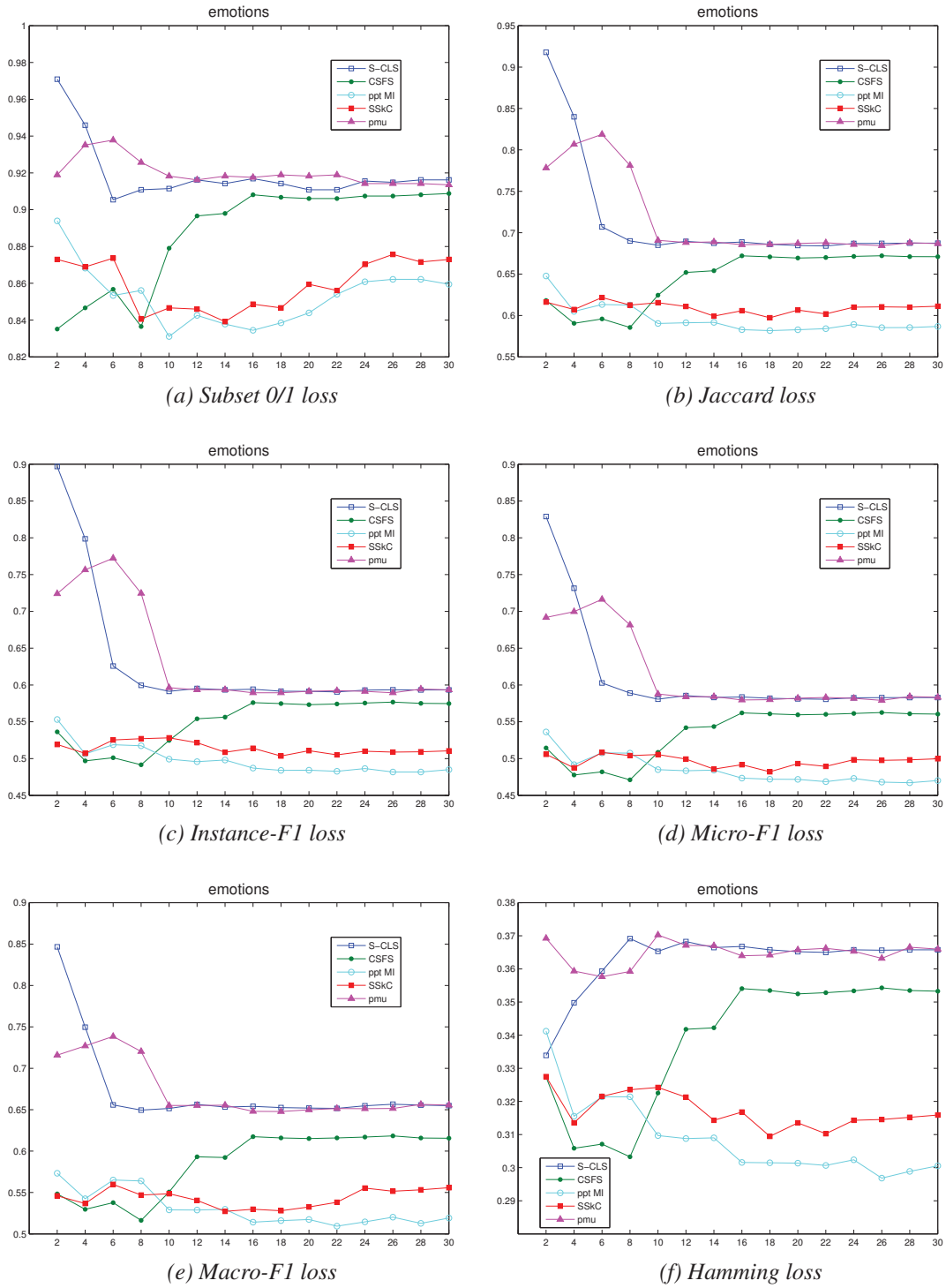
(c) Instance-F1 loss

(d) Micro-F1 loss

(e) Macro-F1 loss

(f) Hamming loss

FIGURE 8.3: Performances metrics averaged over the 25x2 runs vs. different numbers of selected features on Emotions data set.



(a) Subset 0/1 loss

(b) Jaccard loss

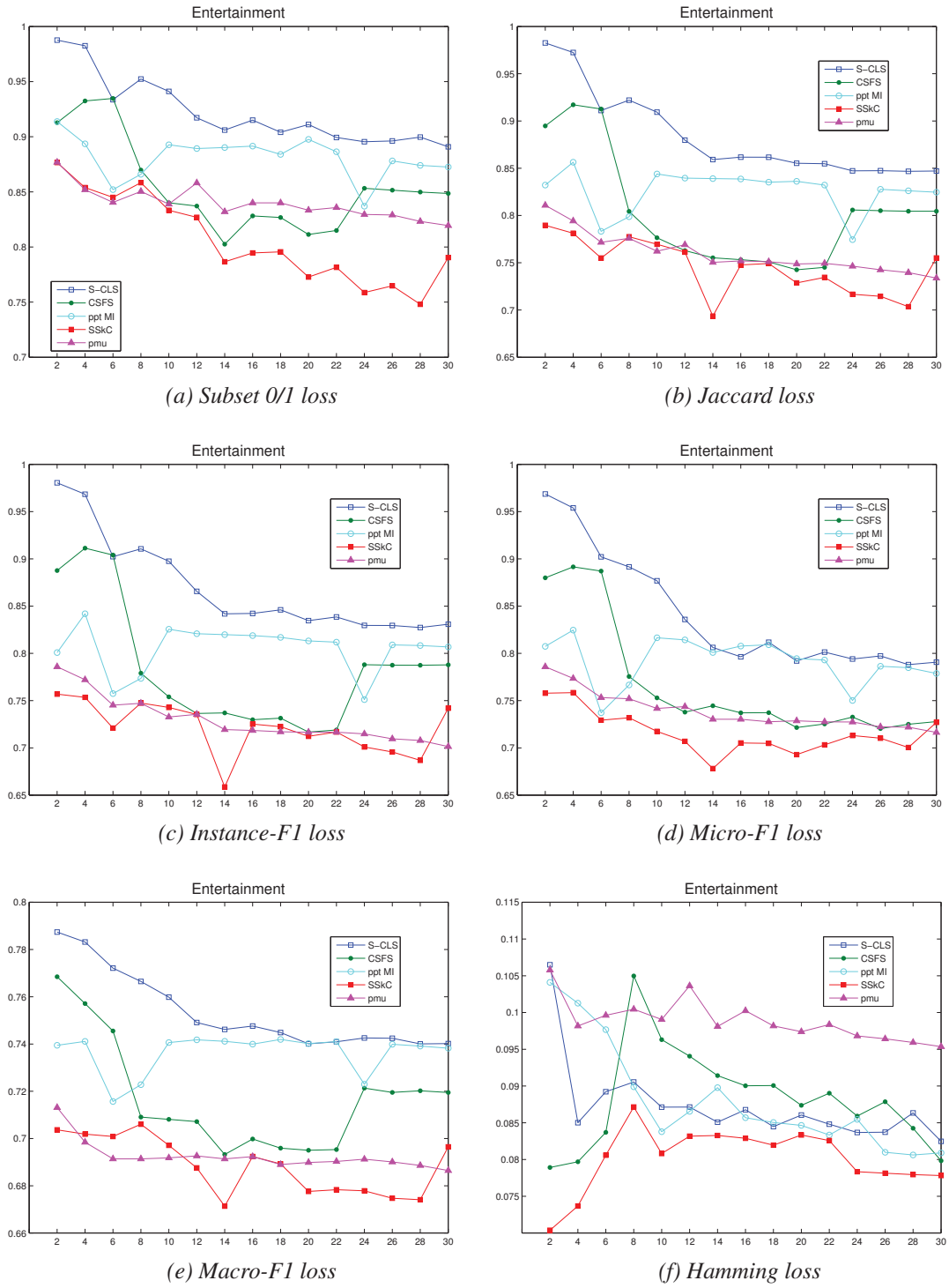
(c) Instance-F1 loss

(d) Micro-F1 loss

(e) Macro-F1 loss

(f) Hamming loss

FIGURE 8.4: Performances metrics averaged over the 25x2 runs vs. different numbers of selected features on Entertainment data set.



(a) Subset 0/1 loss

(b) Jaccard loss

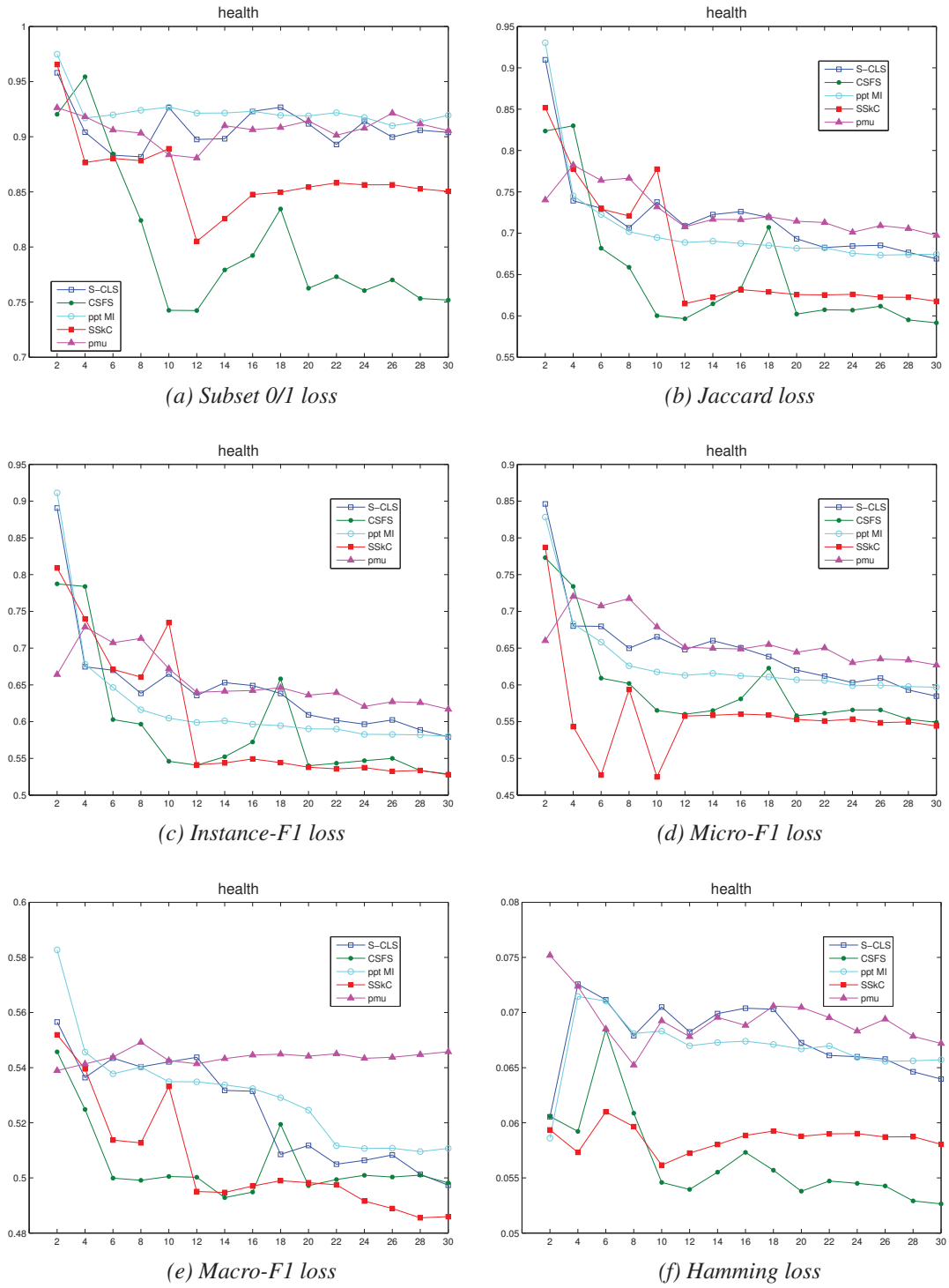
(c) Instance-F1 loss

(d) Micro-F1 loss

(e) Macro-F1 loss

(f) Hamming loss

FIGURE 8.5: Performances metrics averaged over the 25x2 runs vs. different numbers of selected features on Health data set.



(a) Subset 0/1 loss

(b) Jaccard loss

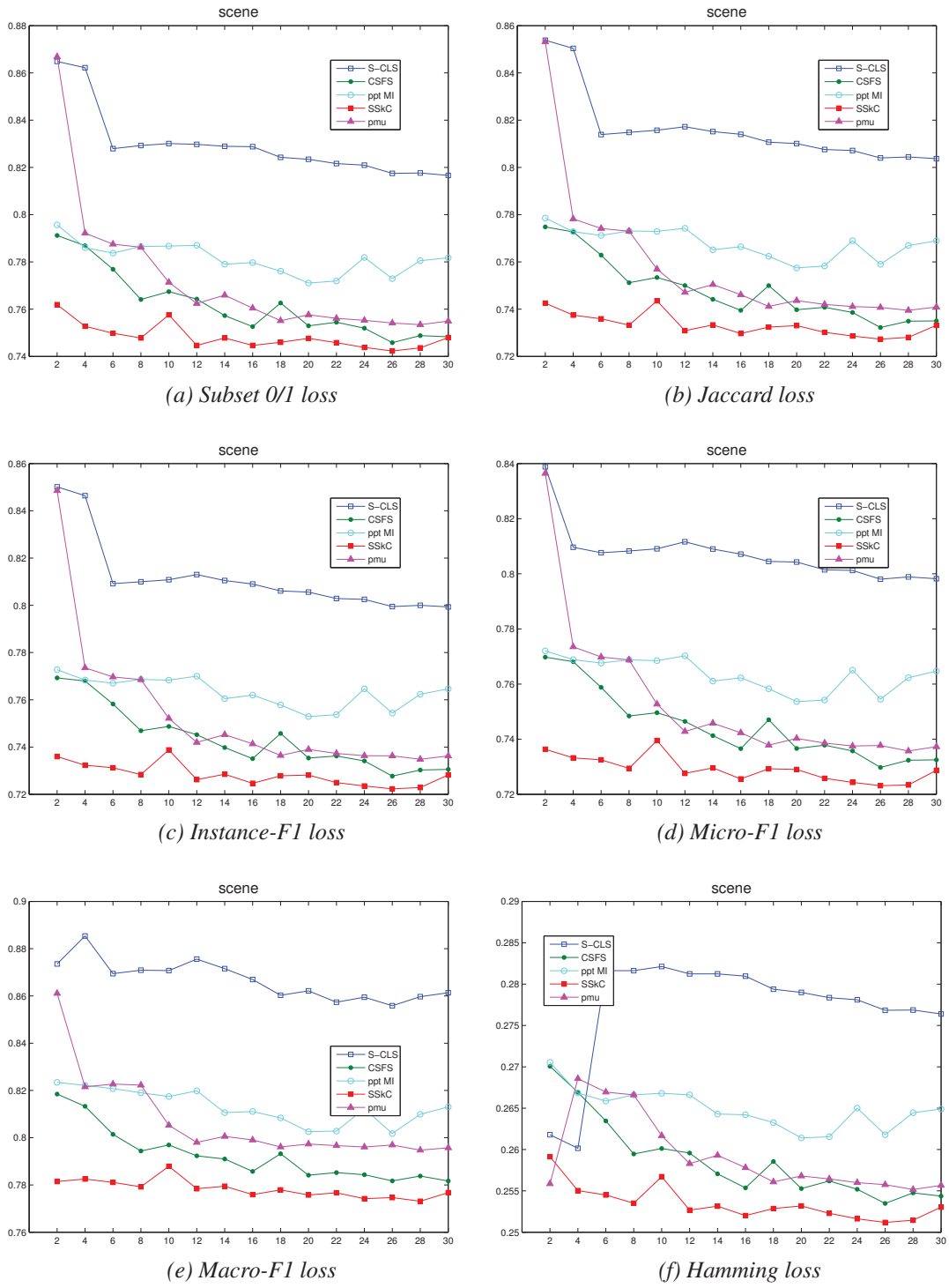
(c) Instance-F1 loss

(d) Micro-F1 loss

(e) Macro-F1 loss

(f) Hamming loss

FIGURE 8.6: Performances metrics averaged over the 25x2 runs vs. different numbers of selected features on Scene data set.



(a) Subset 0/1 loss

(b) Jaccard loss

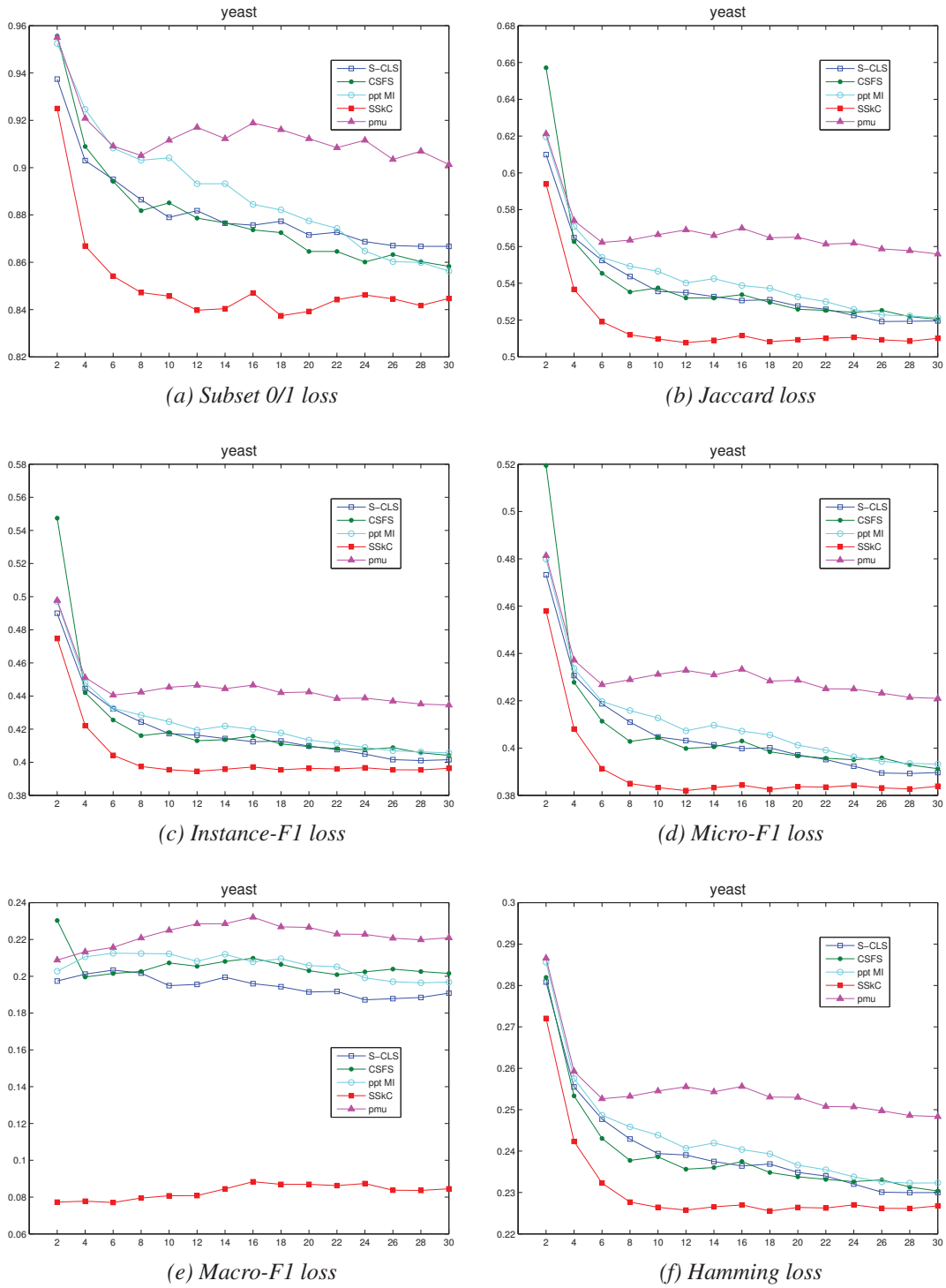
(c) Instance-F1 loss

(d) Micro-F1 loss

(e) Macro-F1 loss

(f) Hamming loss

FIGURE 8.7: Performances metrics averaged over the 25x2 runs vs. different numbers of selected features on Yeast data set.



(a) Subset 0/1 loss

(b) Jaccard loss

(c) Instance-F1 loss

(d) Micro-F1 loss

(e) Macro-F1 loss

(f) Hamming loss

Figures 8.1-8.7 plot the classification performance for each data set in terms of *Subset 0/1 loss*, *Jaccard loss*, *Instance-F1 loss*, *Micro-F1 loss*, *Micro-F1 loss* and *Hamming loss* averaged over the 25x2 runs of the above compared approaches against the 30 most important features (as used in [124]). As expected, we clearly observe that the more features we select, the better

performances we can achieve and that all curves tend to converge as more features are included in the input of the TRAM classifier. Moreover, it may also be observed that, SSkC outperforms the other methods by generally achieving the lowest values over all metrics, except for Emotions (respectively Health) data set for where the *PPT-MI* (respectively *CSFS*) approach performs the best. This indicates the effectiveness of our strategy that includes the loss function consistency throughout different stages of **SSkC** (*i.e.* committee aggregation, instance confidence measure evaluation and feature importance evaluation) to increase dramatically the classification quality in terms of a multi-label performance measure of interest.

The performance of SSkC generally increases swiftly at the beginning (the number of selected feature is small) and slows down at the end. This characteristic suggests that SSkC ranks the features properly and that a classifier can achieve a very good classification accuracy with the top 10 or 12 features while the other methods need more features to achieve comparable results.

TABLE 8.2: *Subset 0/1 loss* averaged over the 30 most important features. The marker '•/∘' indicates that SSkC is significantly better/worse, at a level of significance of 5%.

Data set	SSkC	S-CLS	CSFS	PPT-MI	PMU
Business	.604±.080	.769±.061•	.695±.139•	.752±.075•	.761±.061•
Education	.819±.051	.880±.024•	.836±.050•	.853±.052•	.889±.027•
Emotions	.859±.013	.919±.016•	.887±.028•	.853±.016∘	0.92±.007•
Entertainment	.805±.040	.922±.031•	.854±.041•	.881±.019•	.839±.014•
Health	.863±.035	.908±.019•	.803±.067∘	.923±.014•	.907±.012•
Scene	.748±.005	.829±.014•	.761±.013•	.781±.006•	.771±.029•
Yeast	.850±.021	.881±.018•	.879±.025•	.889±.026•	.913±.012•

TABLE 8.3: *Jaccard loss* averaged over the 30 most important features. The marker '•/∘' indicates that SSkC is significantly better/worse, at a level of significance of 5%.

Data set	SSkC	S-CLS	CSFS	PPT-MI	PMU
Business	.400±.086	.529±.180•	.483±.177•	.495±.151•	.539±.120•
Education	.749±.065	.820±.035•	.776±.061•	.793±.068•	.823±.045•
Emotions	.609±.006	.713±.068•	.645±.033•	.595±.017∘	.716±.050•
Entertainment	.745±.029	.883±.045•	.802±.059•	.825±.022•	.759±.021•
Health	.672±.077	.719±.057•	.650±.079∘	.707±.064•	.725±.026•
Scene	.733±.004	.816±.015•	.747±.013•	.767±.006•	.757±.029•
Yeast	.517±.022	.538±.023•	.540±.033•	.543±.024•	.567±.015•

For the sake of completeness, we also averaged the performances over the different numbers of selected features for each multi-label feature selection algorithm. Tables 8.2-8.7 report the averaged classification performances of the compared algorithms over all considered performance metrics. Algorithms performances are tabulated in terms of averaged values as well as standard

TABLE 8.4: *Instance-F1 loss* averaged over the 30 most important features. The marker '•/∘' indicates that SSkC is significantly better/worse, at a level of significance of 5%.

Data set	SSkC	S-CLS	CSFS	PPT-MI	PMU
Business	.326±.086	.441±.214•	.414±.200•	.407±.177•	.461±.142•
Education	.722±.070	.797±.039•	.752±.065•	.770±.075•	.798±.052•
Emotions	.513±.008	.629±.090•	.550±.032•	.497±.019∘	.632±.070•
Entertainment	.721±.027	.869±.050•	.783±.066•	.805±.025•	.729±.024•
Health	.599±.095	.646±.074•	.592±.085∘	.623±.083•	.654±.035•
Scene	.728±.004	.811±.015•	.743±.013•	.763±.006•	.753±.029•
Yeast	.403±.020	.419±.022•	.423±.035•	.424±.023•	.445±.015•

TABLE 8.5: *Micro-F1 loss* averaged over the 30 most important features. The marker '•/∘' indicates that SSkC is significantly better/worse, at a level of significance of 5%.

Data set	SSkC	S-CLS	CSFS	PPT-MI	PMU
Business	.359±.060	.426±.143•	.421±.168•	.425±.184•	.450±.133•
Education	.727±.061	.800±.037•	.751±.060•	.775±.066•	.801±.049•
Emotions	.496±.007	.610±.071•	.535±.034•	.484±.019∘	.613±.052•
Entertainment	.715±.022	.840±.062•	.766±.063•	.791±.024•	.738±.019•
Health	.560±.069	.649±.062•	.597±.067•	.631±.059•	.660±.031•
Scene	.729±.004	.807±.009•	.744±.012•	.763±.006•	.753±.026•
Yeast	.390±.019	.406±.021•	.409±.031•	.411±.021•	.431±.014•

TABLE 8.6: *Macro-F1 loss* averaged over the 30 most important features. The marker '•/∘' indicates that SSkC is significantly better/worse, at a level of significance of 5%.

Data set	SSkC	S-CLS	CSFS	PPT-MI	PMU
Business	.338±.007	.450±.012•	.448±.020•	.446±.016•	.455±.011•
Education	.433±.016	.466±.007•	.444±.016•	.453±.014•	.456±.010•
Emotions	.543±.010	.672±.054•	.586±.038•	.530±.021∘	.671±.033•
Entertainment	.688±.012	.753±.016•	.717±.023•	.736±.008•	.692±.006•
Health	.505±.020	.524±.019•	.504±.014∘	.529±.019•	.543±.002•
Scene	.778±.003	.866±.008•	.792±.011•	.813±.007•	.806±.018•
Yeast	.483±.003	.494±.005•	.505±.007•	.505±.006•	.522±.006•

deviation on each data set. The lower the value of the considered metric, the better the algorithm performance is. To examine whether the results are statistically significant, paired t-tests were carried out at 5% significance level. The marker '•/∘' suggests that SSkC is statistically superior/inferior to others. Otherwise, a tie is counted and no marker is placed.

Again, SSkC is distinguished from other feature selection methods by achieving, in average, the best performances. This result confirms the ability of our permutation feature importance

TABLE 8.7: *Hamming loss* averaged over the 30 most important features. The marker '•/∘' indicates that SSkC is significantly better/worse, at a level of significance of 5%.

Data set	SSkC	S-CLS	CSFS	PPT-MI	PMU
Business	.035±.004	.043±.003•	.038±.006•	.040±.004•	.044±.003•
Education	.056±.002	.063±.002•	.056±.003•	.058±.004•	.063±.004•
Emotions	.317±.005	.362±.009•	.338±.019•	.308±.011∘	.364±.003•
Entertainment	.080±.004	.087±.005•	.088±.006•	.087±.007•	.098±.002•
Health	.058±.001	.067±.003•	.056±.004∘	.066±.002•	.069±.002•
Scene	.253±.002	.277±.006•	.258±.004•	.264±.002•	.259±.004•
Yeast	.230±.012	.240±.013•	.239±.013•	.243±.013•	.255±.009•

measure to rank the relevant features accurately compared to a fully supervised approach like PMU, due to efficiently exploiting the information from the unlabelled data. Overall, SSkC compares favorably to the other two semi-supervised algorithms that appeared recently in the literature. However, some degradation are reported in performances of SSkC with Emotions data set which is relatively small data set where features have equivalent importance.

8.3 Chapter summary

This Chapter extends our k-labelsets based ensemble method CkMLC [81] to propose and experimentally evaluate a new semi-supervised multi-label feature selection approach based on the ensemble paradigm called SSkC. The proposed method joins ideas from co-training style models and multi-label k-labelsets committee construction in tandem with an inner Random Forest based out-of-bag feature importance evaluation. The three key points are combined in the light of the loss function consistency throughout the different stages of the proposed semi-supervised ensemble approach (*i.e.* committee aggregation, instance confidence measure evaluation and feature importance evaluation). The proposed model differs in the way both labeled and unlabelled out-of-bag instances are used in the learning model and also to evaluate the relevance of the features.

Empirical results on multi-label benchmark data sets indicated that **SSkC** leads to significant improvement over recent state-of-the-art supervised and semi-supervised multi-label feature selection algorithms. The proposed method also shows promise to deal with different multi-label data set domains.

Chapter 9

Conclusion

In this thesis, we addressed the problem of multi-label learning where each instance can be associated with multiple target labels simultaneously. We formulated the multi-label learning as an ensemble learning problem to provide satisfactory solutions for both classification and feature selection tasks. First, we tackled the problem of loss consistency in ensemble multi-label models, especially in the base-classifier combination step. Second, we addressed the multi-label feature selection task, which consists of removing irrelevant and/or redundant features, in both supervised and semi-supervised contexts.

Our main contributions are :

1. A novel strategy to build and aggregate k-labelsets based committee in line with an objective multi-label loss function of interest presented in Chapter 4, competitive and able to achieves good performances compared to the state-of-the-art approaches.

2. A new strategy to combine the base-classifier predictions in conjunction with a new out-of-bag thresholding strategy for ensemble multi-label models. The proposed combination scheme provides a new perspective on the ensemble multi-label mechanisms which investigates the connection between the loss function being optimized by the base classifiers and the loss of the ensemble model. It extends the applicability of ensemble multi-label models with various performance metrics by using (for each specific metric) the adequate combination scheme coupled with an ensemble-based thresholding strategy (if necessary).

3. Three new multi-label feature importance evaluation approaches based on the Random Forest paradigm. These variants optimize different loss metrics depending on the way the label dependence is estimated. Furthermore, we consider the difficult problem of identifying the important features when only a small set of labeled examples is available and propose a new semi-supervised multi-label feature importance evaluation method which

combines ideas from co-training, random k-labelsets ensemble learning and permutation-based out-of-bag feature importance assessment.

In the last few years, dramatic decreases in generalization error in multi-label classification have come about through the growing and combining of an ensemble of diverse multi-label models (e.g. the k-labelsets method and ECC). While diversity is an important factor in this success, care shoud be taken when encouraging diversity in the multi-label context as it may easily hurt the individual performances of multi-label base-classifiers. Furthermore, classical diversity generation methods such as *bagging* - despite being efficient in binary classification - is not well adapted to imbalanced labels distributions in the multi-label context. Thus we found it necessary to investigate the extent to which diversity it is beneficial to the predictive performance of the ensemble, either individually during the training of the base-classifiers or globally when recombining the predictions. The proper manner to enforce diversity was discussed in Chapters 4 and 5, both at the model level and at the combination level, but also in the ensemble feature selection frameworks, either supervised or semi-supervised, presented in Chapter 8.

In addition to this work, we also proposed a novel ensemble multi-label classification method for the specific problem of text categorization [146]. The proposed model termed *Multi Label Rotation Forest*, is based on a combination of two powerful techniques: 1) Rotation Forest [147], one of the most powerful ensemble methods for binary classification problems as shown in extensive experimental studies [148, 149] over a wide range of data sets, and 2) Latent semantic indexing (LSI) an efficient indexing and retrieval method that uses a rank-reduced singular value decomposition (SVD) to identify patterns in the relationships between the words (or terms) and the (latent) concepts. The key idea is to apply the LSI on small random subsets of the vocabulary in order to build a collection of training sets with distinct samples and concept representations. Individual accuracy and diversity within the ensemble are promoted simultaneously. Diversity is promoted through the different splits of the set of words that lead to different orthogonal projections on lower dimensional subspaces, namely the space of concepts. Accuracy is promoted through the underlying latent semantic structure in the text uncovered by LSI. The LSI also reduces noise and other undesirable artifacts of the original space.

An interesting follow-up to our work would be to extend our loss function consistency analysis and feature selection to ensemble multi-target regression problems [150–152]. While a plethora of approaches have been proposed to deal with the challenging task of multi-output regression, the topic of feature selection in multi-output regression is rather unexplored in the literature.

Appendix A

Appendix

A.1 Details of the algorithms performances

This Section provides the tables that present the results of the experiments for each ensemble multi-label method and its variants on 20 multi-label data sets according to the six considered multi-label loss metric : *Subset 0/1 loss*, *Jaccard loss*, *Instance-F1 loss*, *Micro-F1 loss*, *Macro-F1 loss* and *Hamming loss*.

TABLE A.1: Ensemble multi-label variant performances in term of *Subset* 0/1 *loss* (Part 1/2). 'LC' denotes the ***Label Combination*** variant, 'LC' denotes the ***Powerset Combination*** variant, 'LC_{M-T}' denotes the Multi-threshold variant, 'LC_{S-T}' denotes the Single-threshold variant.

Data sets	Variants	EBR	ELP	ECC	RFPCT	VPCME	RAkEL	CkMLC
Arts	LC	.801 ± .005	.830 ± .008	.819 ± .009	.798 ± .006	.942 ± .004	.805 ± .044	.819 ± .045
	PC	.800 ± .005	.648 ± .005	.788 ± .011	.660 ± .006	.940 ± .004	—	—
	LC_{S-T}	.789 ± .002	.702 ± .005	.747 ± .010	.743 ± .008	.882 ± .013	.944 ± .003	.944 ± .006
	LC_{M-T}	.786 ± .005	.701 ± .005	.745 ± .004	.738 ± .007	.879 ± .013	.748 ± .064	.747 ± .067
Birds	LC	.501 ± .011	.520 ± .021	.499 ± .019	.507 ± .016	.528 ± .023	.533 ± .046	.537 ± .043
	PC	.503 ± .014	.495 ± .018	.497 ± .021	.516 ± .016	.528 ± .023	—	—
	LC_{S-T}	.506 ± .014	.489 ± .014	.492 ± .021	.511 ± .016	.511 ± .019	.609 ± .020	.564 ± .025
	LC_{M-T}	.509 ± .017	.488 ± .017	.492 ± .026	.510 ± .016	.516 ± .022	.533 ± .048	.529 ± .039
Business	LC	.469 ± .010	.442 ± .009	.444 ± .009	.462 ± .009	.449 ± .009	.490 ± .095	.478 ± .104
	PC	.462 ± .012	.431 ± .009	.441 ± .010	.447 ± .010	.448 ± .009	—	—
	LC_{S-T}	.468 ± .011	.442 ± .009	.444 ± .009	.453 ± .007	.450 ± .008	.781 ± .047	.674 ± .109
	LC_{M-T}	.458 ± .011	.432 ± .008	.446 ± .011	.452 ± .010	.450 ± .009	.490 ± .096	.473 ± .056
Computers	LC	.672 ± .008	.674 ± .006	.664 ± .005	.648 ± .006	.697 ± .010	.672 ± .008	.671 ± .005
	PC	.672 ± .007	.558 ± .004	.633 ± .005	.565 ± .007	.693 ± .010	—	—
	LC_{S-T}	.665 ± .011	.611 ± .007	.623 ± .004	.620 ± .005	.657 ± .006	.887 ± .006	.876 ± .005
	LC_{M-T}	.650 ± .010	.599 ± .006	.617 ± .006	.627 ± .007	.657 ± .005	.624 ± .008	.622 ± .009
Education	LC	.794 ± .004	.853 ± .003	.835 ± .005	.816 ± .004	.891 ± .007	.800 ± .004	.820 ± .004
	PC	.792 ± .003	.654 ± .005	.780 ± .008	.667 ± .006	.889 ± .007	—	—
	LC_{S-T}	.780 ± .005	.713 ± .003	.759 ± .010	.751 ± .005	.791 ± .010	.957 ± .005	.963 ± .005
	LC_{M-T}	.774 ± .004	.716 ± .004	.742 ± .008	.740 ± .004	.786 ± .011	.728 ± .007	.726 ± .005
Emotions	LC	.720 ± .025	.699 ± .028	.688 ± .017	.689 ± .024	.877 ± .023	.786 ± .021	.721 ± .025
	PC	.720 ± .017	.648 ± .016	.679 ± .013	.663 ± .026	.872 ± .022	—	—
	LC_{S-T}	.727 ± .023	.662 ± .026	.693 ± .020	.695 ± .019	.804 ± .022	.987 ± .007	.910 ± .043
	LC_{M-T}	.717 ± .021	.676 ± .021	.697 ± .019	.692 ± .019	.812 ± .020	.787 ± .026	.772 ± .026
Enron	LC	.886 ± .012	.885 ± .008	.880 ± .006	.875 ± .010	.919 ± .017	.876 ± .007	.887 ± .005
	PC	.890 ± .014	.834 ± .008	.870 ± .009	.854 ± .013	.918 ± .015	—	—
	LC_{S-T}	.885 ± .013	.869 ± .009	.867 ± .009	.874 ± .010	.878 ± .010	.947 ± .006	.940 ± .015
	LC_{M-T}	.855 ± .009	.837 ± .009	.842 ± .009	.853 ± .010	.898 ± .011	.865 ± .009	.869 ± .010
Entertainment	LC	.689 ± .008	.711 ± .004	.699 ± .007	.674 ± .007	.899 ± .014	.668 ± .007	.677 ± .006
	PC	.687 ± .009	.535 ± .004	.662 ± .008	.550 ± .006	.899 ± .014	—	—
	LC_{S-T}	.689 ± .008	.597 ± .003	.648 ± .005	.631 ± .007	.831 ± .035	.875 ± .007	.886 ± .006
	LC_{M-T}	.685 ± .007	.599 ± .008	.646 ± .007	.629 ± .010	.829 ± .035	.623 ± .010	.621 ± .010
Flags	LC	.840 ± .026	.790 ± .019	.811 ± .026	.819 ± .024	.936 ± .028	.803 ± .041	.795 ± .032
	PC	.819 ± .022	.746 ± .030	.783 ± .027	.756 ± .020	.944 ± .024	—	—
	LC_{S-T}	.834 ± .031	.798 ± .026	.811 ± .026	.819 ± .024	.942 ± .019	.987 ± .007	.951 ± .029
	LC_{M-T}	.835 ± .038	.818 ± .028	.836 ± .032	.829 ± .034	.948 ± .030	.825 ± .062	.797 ± .034
Health	LC	.600 ± .007	.596 ± .006	.574 ± .005	.548 ± .004	.741 ± .047	.571 ± .006	.563 ± .005
	PC	.598 ± .008	.490 ± .008	.546 ± .007	.503 ± .005	.733 ± .052	—	—
	LC_{S-T}	.600 ± .007	.545 ± .006	.551 ± .008	.548 ± .004	.663 ± .033	.885 ± .008	.782 ± .098
	LC_{M-T}	.595 ± .006	.517 ± .008	.543 ± .009	.547 ± .006	.663 ± .035	.551 ± .005	.547 ± .005

Complementary of Table A.1
Ensemble multi-label variant performances in term of *Subset* 0/1 *loss* (Part 2/2).
'LC' denotes the ***Label Combination*** variant, 'LC' denotes the ***Powerset Combination*** variant,
'LC_{M-T}' denotes the Multi-threshold variant, 'LC_{S-T}' denotes the Single-threshold variant.

Data sets (↓)	Combination	EBR	ELP	ECC	RFPCT	VPCME	RAkEL	CkMLC
Image	LC	.591 ± .008	.606 ± .010	.585 ± .011	.552 ± .010	.632 ± .014	.650 ± .022	.610 ± .011
	PC	.598 ± .009	.453 ± .014	.561 ± .012	.472 ± .010	.619 ± .018	—	—
	LC_{S-T}	.581 ± .009	.510 ± .011	.546 ± .013	..52 ± .013	.564 ± .010	.896 ± .010	.863 ± .064
	LC_{M-T}	.587 ± .014	.511 ± .015	.547 ± .015	.525 ± .012	.573 ± .014	.644 ± .019	.629 ± .023
Medical	LC	.338 ± .026	.552 ± .022	.640 ± .014	.334 ± .017	.458 ± .024	.314 ± .015	.322 ± .019
	PC	.332 ± .028	.393 ± .018	.630 ± .017	.326 ± .010	.450 ± .025	—	—
	LC_{S-T}	.345 ± .025	.448 ± .012	.442 ± .014	.336 ± .015	.409 ± .012	.475 ± .018	.441 ± .051
	LC_{M-T}	.349 ± .023	.366 ± .009	.385 ± .013	.351 ± .020	.414 ± .013	.317 ± .015	.315 ± .022
Recreation	LC	.755 ± .004	.795 ± .006	.787 ± .004	.749 ± .007	.911 ± .009	.743 ± .006	.754 ± .006
	PC	.756 ± .006	.591 ± .007	.776 ± .005	.605 ± .007	.911 ± .009	—	—
	LC_{S-T}	.754 ± .004	.677 ± .008	.718 ± .005	.687 ± .007	.863 ± .009	.896 ± .006	.898 ± .006
	LC_{M-T}	.761 ± .007	.660 ± .007	.728 ± .007	.687 ± .006	.862 ± .009	.690 ± .006	.689 ± .004
Reference	LC	.637 ± .007	.661 ± .007	.657 ± .005	.636 ± .005	.821 ± .011	.630 ± .005	.635 ± .006
	PC	.638 ± .007	.489 ± .012	.625 ± .005	.497 ± .007	.821 ± .011	—	—
	LC_{S-T}	.621 ± .005	.545 ± .012	.556 ± .009	.575 ± .008	.645 ± .024	.798 ± .008	.792 ± .007
	LC_{M-T}	.596 ± .008	.530 ± .013	.553 ± .007	.559 ± .006	.644 ± .024	.567 ± .005	.561 ± .005
Scene	LC	.461 ± .012	.480 ± .010	.456 ± .011	.431 ± .012	.402 ± .015	.513 ± .018	.481 ± .012
	PC	.467 ± .012	.275 ± .009	.423 ± .011	.302 ± .008	.381 ± .016	—	—
	LC_{S-T}	.422 ± .011	.366 ± .007	.378 ± .007	.374 ± .012	.368 ± .012	.885 ± .013	.847 ± .073
	LC_{M-T}	.432 ± .013	.356 ± .010	.382 ± .009	.375 ± .011	.363 ± .011	.506 ± .018	.494 ± .018
Science	LC	.822 ± .004	.883 ± .005	.871 ± .004	.841 ± .005	.936 ± .004	.817 ± .005	.839 ± .007
	PC	.821 ± .003	.653 ± .008	.850 ± .007	.662 ± .005	.935 ± .004	—	—
	LC_{S-T}	.804 ± .006	.736 ± .005	.772 ± .005	.745 ± .008	.846 ± .009	.940 ± .006	.938 ± .007
	LC_{M-T}	.790 ± .005	.724 ± .007	.765 ± .004	.748 ± .010	.844 ± .009	.736 ± .005	.740 ± .005
Slashdot	LC	.294 ± .010	.293 ± .010	.294 ± .008	.297 ± .012	.360 ± .014	.338 ± .074	.311 ± .074
	PC	.295 ± .009	.293 ± .009	.294 ± .007	.299 ± .011	.360 ± .014	—	—
	LC_{S-T}	.297 ± .012	.293 ± .011	.294 ± .008	.295 ± .012	.364 ± .012	.556 ± .058	.371 ± .018
	LC_{M-T}	.297 ± .011	.293 ± .011	.290 ± .008	.299 ± .012	.362 ± .012	.336 ± .075	.313 ± .030
Social	LC	.535 ± .011	.512 ± .005	.511 ± .005	.503 ± .009	.610 ± .015	.517 ± .009	.501 ± .007
	PC	.534 ± .012	.415 ± .005	.488 ± .005	.425 ± .010	.595 ± .014	—	—
	LC_{S-T}	.535 ± .010	.467 ± .006	.484 ± .006	.479 ± .010	.493 ± .010	.728 ± .011	.745 ± .015
	LC_{M-T}	.522 ± .011	.448 ± .007	.483 ± .006	.481 ± .008	.495 ± .009	.496 ± .008	.491 ± .009
Society	LC	.741 ± .006	.748 ± .004	.739 ± .004	.723 ± .005	.813 ± .016	.753 ± .008	.745 ± .007
	PC	.744 ± .005	.657 ± .008	.725 ± .007	.673 ± .008	.812 ± .016	—	—
	LC_{S-T}	.741 ± .006	.695 ± .008	.707 ± .005	.698 ± .007	.702 ± .010	.937 ± .003	.927 ± .006
	LC_{M-T}	.711 ± .008	.687 ± .008	.704 ± .007	.695 ± .007	.702 ± .010	.702 ± .009	.695 ± .009
Yeast	LC	.846 ± .009	.856 ± .010	.841 ± .006	.822 ± .007	.854 ± .005	.843 ± .011	.854 ± .010
	PC	.875 ± .006	.743 ± .009	.803 ± .011	.753 ± .009	.840 ± .006	—	—
	LC_{S-T}	.844 ± .007	.810 ± .009	.818 ± .007	.823 ± .008	.809 ± .009	.967 ± .005	.941 ± .010
	LC_{M-T}	.829 ± .012	.803 ± .011	.813 ± .010	.807 ± .006	.810 ± .010	.815 ± .008	.811 ± .006

TABLE A.2: Ensemble multi-label variant performances in term of *Jaccard loss* (Part 1/2). 'LC' denotes the ***Label Combination*** variant, 'LC' denotes the ***Powerset Combination*** variant, 'LC_{M-T}' denotes the Multi-threshold variant, 'LC_{S-T}' denotes the Single-threshold variant.

Data sets	Variants	EBR	ELP	ECC	RFPCT	VPCME	RAkEL	CkMLC
Arts	LC	.730 ± .008	.797 ± .009	.777 ± .009	.755 ± .007	.921 ± .005	.866 ± .013	.761 ± .011
	PC	.729 ± .007	.587 ± .006	.745 ± .012	.591 ± .005	.920 ± .005	—	—
	LC_{S-T}	.684 ± .002	.599 ± .006	.650 ± .003	.623 ± .006	.845 ± .014	.647 ± .024	.714 ± .016
	LC_{M-T}	.687 ± .008	.615 ± .008	.637 ± .004	.651 ± .006	.848 ± .014	.658 ± .021	.689 ± .018
Birds	LC	.428 ± .015	.503 ± .022	.452 ± .018	.460 ± .013	.512 ± .023	.481 ± .019	.448 ± .016
	PC	.429 ± .016	.446 ± .024	.451 ± .021	.454 ± .018	.511 ± .024	—	—
	LC_{S-T}	.436 ± .012	.404 ± .015	.406 ± .018	.445 ± .016	.467 ± .020	.454 ± .031	.448 ± .016
	LC_{M-T}	.428 ± .023	.408 ± .015	.400 ± .022	.443 ± .016	.466 ± .016	.428 ± .019	.448 ± .016
Business	LC	.309 ± .006	.297 ± .006	.296 ± .006	.299 ± .006	.302 ± .005	.297 ± .003	.375 ± .025
	PC	.306 ± .008	.292 ± .007	.297 ± .006	.300 ± .007	.302 ± .005	—	—
	LC_{S-T}	.311 ± .008	.289 ± .005	.285 ± .004	.299 ± .006	.294 ± .005	.325 ± .045	.374 ± .012
	LC_{M-T}	.297 ± .008	.282 ± .004	.288 ± .003	.298 ± .007	.295 ± .005	.309 ± .023	.312 ± .052
Computers	LC	.598 ± .008	.617 ± .006	.606 ± .005	.586 ± .006	.630 ± .012	.633 ± .028	.620 ± .006
	PC	.598 ± .008	.486 ± .003	.571 ± .006	.490 ± .007	.626 ± .013	—	—
	LC_{S-T}	.554 ± .014	.515 ± .004	.524 ± .011	.516 ± .006	.560 ± .006	.542 ± .009	.624 ± .004
	LC_{M-T}	.556 ± .010	.503 ± .005	.512 ± .004	.532 ± .006	.560 ± .006	.543 ± .013	.593 ± .006
Education	LC	.734 ± .003	.828 ± .004	.802 ± .004	.781 ± .005	.872 ± .008	.819 ± .011	.789 ± .006
	PC	.732 ± .003	.593 ± .008	.743 ± .008	.601 ± .008	.872 ± .008	—	—
	LC_{S-T}	.671 ± .004	.612 ± .004	.649 ± .009	.626 ± .005	.739 ± .015	.627 ± .005	.714 ± .004
	LC_{M-T}	.670 ± .007	.621 ± .006	.631 ± .005	.642 ± .003	.746 ± .013	.640 ± .012	.677 ± .006
Emotions	LC	.482 ± .018	.488 ± .021	.472 ± .014	.460 ± .022	.681 ± .030	.600 ± .043	.557 ± .008
	PC	.483 ± .017	.415 ± .015	.451 ± .010	.428 ± .026	.671 ± .029	—	—
	LC_{S-T}	.447 ± .012	.425 ± .012	.429 ± .014	.430 ± .021	.537 ± .013	.519 ± .033	.521 ± .008
	LC_{M-T}	.450 ± .013	.425 ± .006	.421 ± .013	.437 ± .019	.545 ± .015	.514 ± .022	.506 ± .019
Enron	LC	.557 ± .016	.613 ± .003	.569 ± .006	.567 ± .011	.648 ± .020	.627 ± .019	.608 ± .003
	PC	.594 ± .014	.595 ± .007	.600 ± .008	.608 ± .006	.647 ± .020	—	—
	LC_{S-T}	.531 ± .011	.530 ± .004	.513 ± .009	.533 ± .003	.573 ± .011	.535 ± .009	.608 ± .008
	LC_{M-T}	.525 ± .011	.524 ± .007	.510 ± .004	.531 ± .008	.574 ± .011	.555 ± .012	.582 ± .005
Entertainment	LC	.625 ± .008	.689 ± .005	.666 ± .008	.634 ± .008	.889 ± .015	.843 ± .024	.643 ± .007
	PC	.625 ± .008	.492 ± .006	.629 ± .009	.496 ± .007	.889 ± .015	—	—
	LC_{S-T}	.600 ± .012	.519 ± .006	.573 ± .006	.527 ± .010	.805 ± .036	.542 ± .014	.645 ± .006
	LC_{M-T}	.603 ± .009	.526 ± .008	.551 ± .008	.551 ± .010	.810 ± .034	.555 ± .008	.601 ± .008
Flags	LC	.415 ± .020	.391 ± .016	.394 ± .017	.398 ± .017	.492 ± .024	.489 ± .035	.434 ± .011
	PC	.422 ± .020	.416 ± .022	.408 ± .017	.416 ± .011	.497 ± .020	—	—
	LC_{S-T}	.394 ± .015	.393 ± .016	.382 ± .019	.394 ± .014	.460 ± .010	.410 ± .024	.434 ± .011
	LC_{M-T}	.389 ± .016	.386 ± .017	.376 ± .010	.387 ± .019	.463 ± .019	.406 ± .032	.434 ± .011
Health	LC	.468 ± .004	.514 ± .005	.477 ± .004	.440 ± .005	.665 ± .051	.606 ± .056	.573 ± .005
	PC	.470 ± .005	.394 ± .007	.449 ± .005	.398 ± .005	.656 ± .057	—	—
	LC_{S-T}	.461 ± .007	.407 ± .006	.409 ± .007	.414 ± .006	.493 ± .032	.431 ± .005	.520 ± .008
	LC_{M-T}	.458 ± .005	.393 ± .007	.415 ± .006	.420 ± .003	.499 ± .035	.425 ± .009	.472 ± .006

Complementary of Table A.2
Ensemble multi-label variant performances in term of *Jaccard loss* (Part 2/2).
'LC' denotes the ***Label Combination*** variant, 'LC' denotes the ***Powerset Combination*** variant,
'LC_{M-T}' denotes the Multi-threshold variant, 'LC_{S-T}' denotes the Single-threshold variant.

Data sets (↓)	Combination	EBR	ELP	ECC	RFPCT	VPCME	RAkEL	CkMLC
Image	LC	.499 ± .007	.540 ± .012	.505 ± .012	.474 ± .010	.563 ± .015	.609 ± .009	.499 ± .032
	PC	.504 ± .008	.365 ± .015	.474 ± .014	.379 ± .012	.548 ± .019	—	—
	LC_{S-T}	.443 ± .010	.385 ± .010	.422 ± .016	.389 ± .011	.430 ± .012	.609 ± .009	.499 ± .032
	LC_{M-T}	.451 ± .014	.392 ± .012	.416 ± .009	.407 ± .018	.436 ± .014	.609 ± .009	.499 ± .032
Medical	LC	.249 ± .023	.494 ± .024	.580 ± .019	.258 ± .019	.393 ± .025	.322 ± .023	.284 ± .010
	PC	.247 ± .024	.311 ± .014	.568 ± .019	.254 ± .014	.384 ± .026	—	—
	LC_{S-T}	.243 ± .016	.338 ± .018	.276 ± .017	.249 ± .016	.309 ± .012	.237 ± .019	.268 ± .014
	LC_{M-T}	.243 ± .019	.267 ± .015	.328 ± .010	.261 ± .013	.307 ± .010	.236 ± .018	.246 ± .019
Recreation	LC	.705 ± .006	.776 ± .006	.765 ± .004	.718 ± .008	.902 ± .009	.852 ± .012	.729 ± .005
	PC	.705 ± .006	.547 ± .007	.753 ± .006	.554 ± .006	.902 ± .009	—	—
	LC_{S-T}	.673 ± .004	.563 ± .007	.651 ± .003	.604 ± .006	.844 ± .010	.618 ± .005	.682 ± .006
	LC_{M-T}	.684 ± .004	.587 ± .009	.634 ± .006	.610 ± .009	.846 ± .010	.625 ± .007	.645 ± .009
Reference	LC	.589 ± .007	.640 ± .007	.634 ± .005	.608 ± .005	.808 ± .011	.711 ± .047	.610 ± .005
	PC	.589 ± .007	.444 ± .012	.598 ± .005	.451 ± .007	.807 ± .011	—	—
	LC_{S-T}	.546 ± .006	.472 ± .010	.486 ± .009	.485 ± .007	.590 ± .029	.499 ± .006	.570 ± .005
	LC_{M-T}	.521 ± .012	.466 ± .014	.488 ± .007	.493 ± .008	.592 ± .029	.496 ± .007	.542 ± .005
Scene	LC	.426 ± .012	.459 ± .010	.431 ± .010	.404 ± .010	.367 ± .014	.542 ± .007	.317 ± .025
	PC	.431 ± .011	.243 ± .008	.398 ± .011	.271 ± .008	.348 ± .016	—	—
	LC_{S-T}	.348 ± .007	.284 ± .005	.312 ± .007	.315 ± .005	.284 ± .007	.542 ± .007	.317 ± .025
	LC_{M-T}	.355 ± .010	.297 ± .011	.310 ± .009	.314 ± .011	.282 ± .008	.542 ± .007	.317 ± .025
Science	LC	.770 ± .004	.872 ± .006	.851 ± .005	.815 ± .005	.925 ± .005	.866 ± .019	.817 ± .007
	PC	.771 ± .004	.611 ± .008	.829 ± .008	.612 ± .005	.925 ± .005	—	—
	LC_{S-T}	.704 ± .004	.634 ± .008	.686 ± .006	.662 ± .006	.814 ± .011	.668 ± .008	.712 ± .005
	LC_{M-T}	.712 ± .006	.655 ± .008	.672 ± .007	.670 ± .010	.817 ± .012	.668 ± .007	.688 ± .009
Slashdot	LC	.231 ± .007	.232 ± .006	.233 ± .004	.233 ± .008	.296 ± .010	.291 ± .012	.284 ± .020
	PC	.233 ± .006	.232 ± .005	.233 ± .004	.235 ± .007	.296 ± .010	—	—
	LC_{S-T}	.236 ± .007	.232 ± .006	.230 ± .005	.234 ± .007	.299 ± .007	.267 ± .046	.284 ± .020
	LC_{M-T}	.235 ± .007	.231 ± .007	.234 ± .005	.236 ± .008	.299 ± .007	.258 ± .011	.284 ± .020
Social	LC	.465 ± .010	.482 ± .004	.476 ± .007	.459 ± .009	.583 ± .016	.593 ± .040	.506 ± .010
	PC	.466 ± .011	.368 ± .006	.451 ± .006	.374 ± .009	.567 ± .015	—	—
	LC_{S-T}	.454 ± .007	.399 ± .009	.403 ± .004	.406 ± .006	.413 ± .007	.420 ± .007	.506 ± .010
	LC_{M-T}	.438 ± .006	.381 ± .007	.402 ± .006	.410 ± .008	.414 ± .008	.420 ± .010	.506 ± .010
Society	LC	.656 ± .007	.694 ± .004	.679 ± .005	.650 ± .007	.773 ± .019	.757 ± .039	.690 ± .009
	PC	.659 ± .007	.562 ± .009	.656 ± .008	.576 ± .008	.771 ± .020	—	—
	LC_{S-T}	.623 ± .007	.579 ± .006	.592 ± .008	.584 ± .007	.607 ± .010	.591 ± .008	.696 ± .006
	LC_{M-T}	.605 ± .012	.576 ± .007	.587 ± .008	.585 ± .009	.609 ± .009	.588 ± .007	.650 ± .008
Yeast	LC	.496 ± .005	.525 ± .006	.503 ± .005	.485 ± .005	.520 ± .004	.594 ± .005	.491 ± .005
	PC	.541 ± .007	.469 ± .008	.503 ± .010	.470 ± .008	.517 ± .005	—	—
	LC_{S-T}	.469 ± .007	.457 ± .004	.452 ± .005	.452 ± .005	.454 ± .004	.594 ± .005	.491 ± .005
	LC_{M-T}	.460 ± .004	.454 ± .005	.453 ± .004	.455 ± .005	.454 ± .005	.594 ± .005	.491 ± .005

TABLE A.3: Ensemble multi-label variant performances in term of *Instance-F1 loss* (Part 1/2). 'LC' denotes the ***Label Combination*** variant, 'LC' denotes the ***Powerset Combination*** variant, 'LC_{M-T}' denotes the Multi-threshold variant, 'LC_{S-T}' denotes the Single-threshold variant.

Data sets	Variants	EBR	ELP	ECC	RFPCT	VPCME	RAkEL	CkMLC
Arts	LC	.703 ± .009	.785 ± .009	.762 ± .010	.739 ± .007	.913 ± .005	.855 ± .014	.737 ± .038
	PC	.703 ± .008	.563 ± .006	.728 ± .012	.563 ± .005	.913 ± .005	—	—
	LC_{S-T}	.620 ± .005	.573 ± .006	.603 ± .003	.604 ± .006	.836 ± .015	.623 ± .008	.622 ± .044
	LC_{M-T}	.636 ± .007	.536 ± .006	.570 ± .003	.567 ± .008	.832 ± .015	.624 ± .006	.580 ± .004
Birds	LC	.401 ± .018	.496 ± .023	.433 ± .017	.442 ± .013	.505 ± .024	.467 ± .019	.443 ± .015
	PC	.402 ± .017	.427 ± .028	.432 ± .021	.432 ± .020	.504 ± .025	—	—
	LC_{S-T}	.397 ± .017	.372 ± .014	.371 ± .019	.415 ± .018	.447 ± .015	.445 ± .021	.440 ± .015
	LC_{M-T}	.393 ± .026	.362 ± .016	.361 ± .017	.410 ± .019	.448 ± .020	.446 ± .021	.414 ± .023
Business	LC	.252 ± .006	.244 ± .006	.243 ± .005	.242 ± .005	.248 ± .004	.241 ± .003	.257 ± .033
	PC	.251 ± .007	.240 ± .006	.245 ± .005	.247 ± .006	.248 ± .004	—	—
	LC_{S-T}	.248 ± .006	.227 ± .004	.228 ± .004	.239 ± .005	.239 ± .005	.364 ± .011	.257 ± .033
	LC_{M-T}	.238 ± .007	.238 ± .007	.231 ± .003	.242 ± .005	.238 ± .004	.358 ± .025	.257 ± .033
Computers	LC	.570 ± .009	.596 ± .007	.584 ± .005	.562 ± .006	.604 ± .013	.607 ± .030	.600 ± .007
	PC	.570 ± .008	.458 ± .003	.546 ± .007	.462 ± .008	.600 ± .014	—	—
	LC_{S-T}	.500 ± .016	.453 ± .007	.472 ± .013	.474 ± .012	.526 ± .008	.533 ± .004	.513 ± .004
	LC_{M-T}	.494 ± .013	.468 ± .009	.456 ± .006	.458 ± .006	.523 ± .007	.533 ± .004	.499 ± .004
Education	LC	.713 ± .004	.820 ± .004	.790 ± .004	.769 ± .006	.866 ± .008	.809 ± .012	.778 ± .006
	PC	.711 ± .004	.571 ± .009	.730 ± .008	.577 ± .008	.865 ± .009	—	—
	LC_{S-T}	.606 ± .008	.573 ± .008	.597 ± .012	.592 ± .004	.730 ± .014	.612 ± .003	.596 ± .004
	LC_{M-T}	.620 ± .008	.542 ± .019	.560 ± .003	.566 ± .005	.722 ± .017	.612 ± .003	.570 ± .006
Emotions	LC	.404 ± .017	.419 ± .022	.401 ± .016	.384 ± .023	.611 ± .032	.518 ± .051	.436 ± .019
	PC	.405 ± .017	.337 ± .016	.376 ± .012	.348 ± .026	.599 ± .033	—	—
	LC_{S-T}	.352 ± .016	.337 ± .012	.342 ± .010	.345 ± .020	.454 ± .016	.527 ± .008	.414 ± .006
	LC_{M-T}	.356 ± .012	.334 ± .012	.324 ± .011	.336 ± .010	.439 ± .013	.518 ± .028	.390 ± .009
Enron	LC	.444 ± .017	.509 ± .005	.459 ± .007	.457 ± .012	.551 ± .021	.528 ± .020	.501 ± .004
	PC	.484 ± .015	.503 ± .008	.498 ± .008	.512 ± .006	.550 ± .021	—	—
	LC_{S-T}	.415 ± .015	.405 ± .007	.396 ± .005	.410 ± .004	.471 ± .011	.501 ± .007	.471 ± .007
	LC_{M-T}	.404 ± .012	.411 ± .003	.391 ± .004	.410 ± .003	.466 ± .011	.501 ± .007	.451 ± .006
Entertainment	LC	.602 ± .008	.681 ± .005	.655 ± .009	.620 ± .008	.886 ± .015	.837 ± .024	.630 ± .007
	PC	.602 ± .009	.476 ± .007	.617 ± .009	.477 ± .007	.885 ± .015	—	—
	LC_{S-T}	.557 ± .015	.489 ± .008	.533 ± .010	.517 ± .010	.803 ± .035	.545 ± .004	.533 ± .006
	LC_{M-T}	.561 ± .007	.498 ± .004	.500 ± .009	.484 ± .009	.797 ± .036	.545 ± .004	.503 ± .008
Flags	LC	.300 ± .021	.280 ± .015	.281 ± .014	.287 ± .016	.357 ± .021	.353 ± .036	.292 ± .019
	PC	.309 ± .020	.311 ± .023	.298 ± .015	.308 ± .012	.360 ± .018	—	—
	LC_{S-T}	.269 ± .016	.264 ± .013	.264 ± .011	.274 ± .016	.325 ± .021	.353 ± .009	.290 ± .009
	LC_{M-T}	.270 ± .016	.264 ± .009	.258 ± .011	.273 ± .015	.317 ± .010	.353 ± .009	.281 ± .009
Health	LC	.421 ± .004	.484 ± .005	.442 ± .004	.401 ± .005	.637 ± .053	.572 ± .057	.439 ± .006
	PC	.424 ± .004	.358 ± .006	.414 ± .005	.360 ± .005	.628 ± .059	—	—
	LC_{S-T}	.388 ± .005	.339 ± .007	.355 ± .006	.363 ± .006	.445 ± .044	.469 ± .005	.447 ± .005
	LC_{M-T}	.393 ± .008	.348 ± .005	.346 ± .007	.359 ± .006	.432 ± .034	.469 ± .005	.423 ± .003

Complementary of Table A.3
Ensemble multi-label variant performances in term of *Instance-F1 loss* (Part 2/2).
'LC' denotes the ***Label Combination*** variant, 'LC' denotes the ***Powerset Combination*** variant,
'LC_{M-T}' denotes the Multi-threshold variant, 'LC_{S-T}' denotes the Single-threshold variant.

Data sets (↓)	Combination	EBR	ELP	ECC	RFPCT	VPCME	RAkEL	CkMLC
Image	LC	.468 ± .008	.517 ± .013	.477 ± .013	.448 ± .011	.540 ± .015	.469 ± .035	.519 ± .010
	PC	.472 ± .008	.335 ± .016	.445 ± .014	.348 ± .013	.523 ± .020	—	—
	LC_{S-T}	.384 ± .011	.339 ± .009	.365 ± .016	.354 ± .018	.391 ± .018	.606 ± .010	.470 ± .008
	LC_{M-T}	.394 ± .019	.320 ± .023	.335 ± .005	.333 ± .009	.378 ± .014	.598 ± .034	.433 ± .008
Medical	LC	.219 ± .023	.474 ± .025	.559 ± .022	.232 ± .020	.371 ± .025	.295 ± .024	.220 ± .019
	PC	.219 ± .023	.284 ± .013	.547 ± .020	.230 ± .015	.362 ± .026	—	—
	LC_{S-T}	.214 ± .025	.225 ± .021	.230 ± .018	.227 ± .016	.277 ± .011	.227 ± .016	.218 ± .011
	LC_{M-T}	.204 ± .021	.282 ± .012	.266 ± .010	.215 ± .015	.272 ± .012	.227 ± .016	.209 ± .013
Recreation	LC	.686 ± .007	.769 ± .006	.757 ± .004	.707 ± .008	.898 ± .009	.846 ± .012	.719 ± .005
	PC	.686 ± .007	.530 ± .007	.745 ± .006	.535 ± .006	.898 ± .010	—	—
	LC_{S-T}	.631 ± .016	.553 ± .010	.608 ± .005	.577 ± .010	.840 ± .011	.597 ± .004	.579 ± .005
	LC_{M-T}	.641 ± .006	.518 ± .007	.573 ± .005	.540 ± .004	.837 ± .010	.597 ± .004	.554 ± .008
Reference	LC	.572 ± .007	.633 ± .007	.627 ± .005	.598 ± .005	.803 ± .011	.702 ± .048	.601 ± .005
	PC	.571 ± .007	.429 ± .012	.589 ± .005	.435 ± .008	.803 ± .011	—	—
	LC_{S-T}	.517 ± .014	.435 ± .012	.453 ± .008	.460 ± .008	.574 ± .031	.490 ± .003	.477 ± .004
	LC_{M-T}	.486 ± .007	.462 ± .023	.437 ± .008	.448 ± .007	.571 ± .030	.490 ± .003	.462 ± .004
Scene	LC	.415 ± .012	.452 ± .010	.423 ± .010	.396 ± .010	.355 ± .014	.300 ± .028	.449 ± .013
	PC	.419 ± .010	.233 ± .007	.389 ± .011	.260 ± .009	.336 ± .016	—	—
	LC_{S-T}	.324 ± .008	.267 ± .012	.284 ± .004	.285 ± .015	.253 ± .011	.681 ± .009	.404 ± .006
	LC_{M-T}	.319 ± .014	.262 ± .036	.260 ± .009	.260 ± .004	.245 ± .008	.664 ± .056	.372 ± .008
Science	LC	.751 ± .005	.868 ± .007	.844 ± .005	.805 ± .005	.921 ± .005	.859 ± .020	.809 ± .008
	PC	.752 ± .005	.596 ± .009	.822 ± .008	.594 ± .005	.922 ± .005	—	—
	LC_{S-T}	.655 ± .005	.614 ± .009	.646 ± .009	.630 ± .008	.806 ± .013	.623 ± .006	.611 ± .005
	LC_{M-T}	.676 ± .008	.587 ± .008	.610 ± .004	.589 ± .005	.803 ± .012	.623 ± .006	.596 ± .008
Slashdot	LC	.210 ± .006	.211 ± .004	.212 ± .003	.211 ± .007	.274 ± .008	.269 ± .010	.217 ± .038
	PC	.212 ± .006	.211 ± .004	.212 ± .003	.212 ± .006	.274 ± .008	—	—
	LC_{S-T}	.212 ± .007	.210 ± .006	.210 ± .004	.214 ± .006	.276 ± .005	.352 ± .015	.217 ± .038
	LC_{M-T}	.214 ± .006	.212 ± .005	.213 ± .004	.211 ± .006	.277 ± .005	.345 ± .032	.217 ± .038
Social	LC	.440 ± .011	.471 ± .004	.463 ± .008	.444 ± .009	.573 ± .017	.583 ± .042	.449 ± .008
	PC	.441 ± .011	.350 ± .006	.438 ± .006	.355 ± .009	.557 ± .015	—	—
	LC_{S-T}	.417 ± .015	.350 ± .006	.371 ± .009	.377 ± .008	.387 ± .009	.417 ± .004	.409 ± .008
	LC_{M-T}	.399 ± .008	.362 ± .008	.367 ± .006	.362 ± .007	.384 ± .008	.417 ± .004	.390 ± .005
Society	LC	.623 ± .008	.673 ± .005	.655 ± .005	.622 ± .008	.757 ± .021	.739 ± .043	.668 ± .010
	PC	.626 ± .007	.525 ± .009	.630 ± .008	.538 ± .007	.755 ± .021	—	—
	LC_{S-T}	.566 ± .011	.523 ± .007	.537 ± .012	.532 ± .010	.572 ± .013	.614 ± .006	.585 ± .006
	LC_{M-T}	.551 ± .013	.516 ± .013	.533 ± .008	.523 ± .007	.570 ± .011	.614 ± .006	.554 ± .005
Yeast	LC	.388 ± .005	.417 ± .005	.396 ± .004	.379 ± .004	.412 ± .003	.385 ± .004	.411 ± .005
	PC	.431 ± .007	.376 ± .007	.404 ± .009	.376 ± .009	.411 ± .004	—	—
	LC_{S-T}	.352 ± .003	.343 ± .005	.340 ± .005	.343 ± .004	.349 ± .004	.464 ± .004	.445 ± .004
	LC_{M-T}	.347 ± .003	.345 ± .009	.343 ± .005	.342 ± .005	.347 ± .003	.464 ± .004	.414 ± .005

TABLE A.4: Ensemble multi-label variant performances in term of *Micro-F1 loss*(Part 1/2). 'LC' denotes the ***Label Combination*** variant, 'LC' denotes the ***Powerset Combination*** variant, 'LC_{M-T}' denotes the Multi-threshold variant, 'LC_{S-T}' denotes the Single-threshold variant.

Data sets	Variants	EBR	ELP	ECC	RFPCT	VPCME	RAkEL	CkMLC
Arts	LC	.649 ± .006	.726 ± .009	.698 ± .007	.607 ± .006	.870 ± .008	.662 ± .009	.681 ± .011
	PC	.653 ± .006	.595 ± .006	.675 ± .009	.595 ± .006	.870 ± .007	—	—
	LC_{S-T}	.632 ± .006	.613 ± .036	.598 ± .004	.673 ± .006	.769 ± .013	.662 ± .010	.634 ± .015
	LC_{M-T}	.616 ± .006	.572 ± .008	.579 ± .004	.584 ± .005	.768 ± .013	.588 ± .032	.582 ± .022
Birds	LC	.611 ± .027	.869 ± .027	.706 ± .013	.613 ± .025	.886 ± .038	.655 ± .038	.699 ± .054
	PC	.612 ± .023	.697 ± .043	.704 ± .021	.683 ± .043	.884 ± .037	—	—
	LC_{S-T}	.578 ± .023	.597 ± .035	.548 ± .016	.708 ± .033	.716 ± .029	.635 ± .018	.627 ± .028
	LC_{M-T}	.566 ± .021	.571 ± .027	.539 ± .018	.594 ± .024	.732 ± .041	.610 ± .074	.587 ± .034
Business	LC	.293 ± .007	.296 ± .007	.291 ± .006	.327 ± .010	.301 ± .006	.296 ± .026	.296 ± .028
	PC	.294 ± .008	.292 ± .008	.296 ± .006	.298 ± .006	.301 ± .005	—	—
	LC_{S-T}	.325 ± .012	.362 ± .007	.288 ± .008	.284 ± .006	.289 ± .006	.399 ± .015	.370 ± .027
	LC_{M-T}	.343 ± .036	.279 ± .004	.273 ± .004	.285 ± .006	.289 ± .005	.285 ± .005	.293 ± .030
Computers	LC	.530 ± .009	.552 ± .008	.533 ± .007	.518 ± .012	.577 ± .010	.551 ± .007	.551 ± .006
	PC	.532 ± .008	.489 ± .004	.516 ± .006	.488 ± .008	.575 ± .011	—	—
	LC_{S-T}	.536 ± .008	.544 ± .002	.488 ± .004	.514 ± .007	.523 ± .005	.576 ± .005	.552 ± .003
	LC_{M-T}	.515 ± .011	.478 ± .005	.473 ± .004	.482 ± .009	.521 ± .005	.488 ± .006	.484 ± .007
Education	LC	.632 ± .006	.728 ± .004	.701 ± .005	.575 ± .004	.784 ± .011	.656 ± .006	.681 ± .009
	PC	.633 ± .006	.572 ± .009	.651 ± .007	.583 ± .009	.783 ± .011	—	—
	LC_{S-T}	.597 ± .007	.566 ± .039	.558 ± .003	.673 ± .009	.636 ± .014	.638 ± .002	.615 ± .003
	LC_{M-T}	.587 ± .005	.536 ± .006	.550 ± .005	.558 ± .006	.635 ± .014	.543 ± .003	.541 ± .004
Emotions	LC	.342 ± .017	.346 ± .014	.333 ± .013	.320 ± .020	.552 ± .026	.425 ± .016	.363 ± .016
	PC	.344 ± .016	.312 ± .014	.324 ± .011	.323 ± .024	.545 ± .026	—	—
	LC_{S-T}	.331 ± .006	.309 ± .010	.311 ± .010	.332 ± .020	.417 ± .017	.516 ± .007	.413 ± .007
	LC_{M-T}	.320 ± .009	.312 ± .016	.302 ± .014	.313 ± .015	.415 ± .015	.380 ± .020	.366 ± .017
Enron	LC	.423 ± .009	.493 ± .005	.442 ± .005	.457 ± .005	.503 ± .009	.487 ± .005	.491 ± .003
	PC	.473 ± .008	.504 ± .008	.492 ± .006	.524 ± .004	.501 ± .009	—	—
	LC_{S-T}	.460 ± .007	.466 ± .034	.433 ± .007	.447 ± .006	.435 ± .006	.517 ± .007	.482 ± .005
	LC_{M-T}	.433 ± .017	.417 ± .008	.389 ± .003	.411 ± .002	.434 ± .006	.405 ± .004	.396 ± .005
Entertainment	LC	.542 ± .008	.611 ± .005	.571 ± .008	.511 ± .011	.825 ± .020	.551 ± .007	.568 ± .005
	PC	.544 ± .008	.501 ± .006	.550 ± .006	.500 ± .007	.825 ± .020	—	—
	LC_{S-T}	.543 ± .007	.490 ± .006	.490 ± .008	.546 ± .007	.716 ± .035	.602 ± .006	.579 ± .007
	LC_{M-T}	.528 ± .012	.467 ± .009	.483 ± .010	.492 ± .007	.714 ± .035	.490 ± .008	.487 ± .008
Flags	LC	.266 ± .016	.253 ± .012	.254 ± .013	.258 ± .016	.346 ± .022	.276 ± .025	.260 ± .017
	PC	.276 ± .016	.282 ± .019	.272 ± .012	.278 ± .010	.348 ± .018	—	—
	LC_{S-T}	.247 ± .012	.254 ± .018	.246 ± .014	.255 ± .013	.302 ± .013	.336 ± .010	.272 ± .008
	LC_{M-T}	.248 ± .012	.251 ± .012	.240 ± .011	.250 ± .014	.303 ± .009	.256 ± .014	.249 ± .016
Health	LC	.402 ± .004	.442 ± .004	.409 ± .003	.390 ± .006	.569 ± .049	.407 ± .006	.408 ± .005
	PC	.405 ± .004	.375 ± .004	.402 ± .003	.382 ± .004	.563 ± .053	—	—
	LC_{S-T}	.409 ± .005	.386 ± .006	.359 ± .006	.388 ± .004	.442 ± .041	.508 ± .003	.478 ± .005
	LC_{M-T}	.400 ± .009	.349 ± .004	.364 ± .005	.373 ± .005	.436 ± .030	.370 ± .006	.369 ± .007

Complementary of Table A.4
Ensemble multi-label variant performances in term of *Micro-F1 loss* (Part 2/2).
'LC' denotes the ***Label Combination*** variant, 'LC' denotes the ***Powerset Combination*** variant,
'LC_{M-T}' denotes the Multi-threshold variant, 'LC_{S-T}' denotes the Single-threshold variant.

Data sets (↓)	Combination	EBR	ELP	ECC	RFPCT	VPCME	RAkEL	CkMLC
Image	LC	.395 ± .008	.426 ± .011	.396 ± .010	.339 ± .013	.454 ± .012	.462 ± .019	.428 ± .009
	PC	.401 ± .009	.344 ± .015	.383 ± .012	.354 ± .011	.445 ± .015	—	—
	LC_{S-T}	.363 ± .012	.332 ± .011	.342 ± .011	.383 ± .008	.373 ± .012	.599 ± .009	.493 ± .006
	LC_{M-T}	.360 ± .012	.330 ± .011	.344 ± .009	.336 ± .008	.373 ± .011	.428 ± .013	.413 ± .013
Medical	LC	.191 ± .016	.369 ± .020	.429 ± .020	.258 ± .014	.294 ± .019	.194 ± .012	.196 ± .014
	PC	.192 ± .017	.287 ± .016	.422 ± .017	.235 ± .015	.291 ± .019	—	—
	LC_{S-T}	.218 ± .018	.366 ± .047	.217 ± .012	.219 ± .018	.253 ± .009	.255 ± .015	.237 ± .018
	LC_{M-T}	.228 ± .017	.222 ± .014	.257 ± .008	.217 ± .016	.258 ± .010	.194 ± .013	.189 ± .015
Recreation	LC	.634 ± .007	.704 ± .008	.689 ± .005	.587 ± .007	.851 ± .013	.638 ± .007	.656 ± .005
	PC	.637 ± .007	.568 ± .006	.678 ± .007	.575 ± .004	.850 ± .014	—	—
	LC_{S-T}	.624 ± .005	.548 ± .007	.588 ± .007	.643 ± .007	.775 ± .011	.658 ± .003	.631 ± .005
	LC_{M-T}	.619 ± .007	.545 ± .005	.582 ± .005	.568 ± .005	.775 ± .011	.561 ± .005	.557 ± .006
Reference	LC	.502 ± .006	.535 ± .006	.523 ± .005	.483 ± .008	.711 ± .013	.500 ± .004	.507 ± .004
	PC	.504 ± .007	.442 ± .012	.500 ± .006	.453 ± .007	.711 ± .012	—	—
	LC_{S-T}	.510 ± .010	.459 ± .031	.448 ± .007	.511 ± .006	.524 ± .013	.559 ± .003	.533 ± .004
	LC_{M-T}	.492 ± .012	.425 ± .010	.441 ± .010	.454 ± .008	.523 ± .013	.446 ± .005	.442 ± .005
Scene	LC	.311 ± .010	.332 ± .007	.308 ± .007	.268 ± .006	.285 ± .010	.376 ± .020	.333 ± .010
	PC	.316 ± .009	.239 ± .007	.292 ± .008	.265 ± .008	.277 ± .010	—	—
	LC_{S-T}	.276 ± .011	.250 ± .004	.244 ± .007	.306 ± .008	.252 ± .006	.680 ± .008	.444 ± .007
	LC_{M-T}	.269 ± .007	.246 ± .010	.247 ± .004	.265 ± .008	.254 ± .007	.356 ± .018	.347 ± .015
Science	LC	.690 ± .003	.817 ± .010	.784 ± .007	.621 ± .010	.880 ± .008	.713 ± .008	.742 ± .010
	PC	.694 ± .003	.624 ± .008	.763 ± .009	.618 ± .005	.880 ± .008	—	—
	LC_{S-T}	.654 ± .006	.643 ± .004	.606 ± .007	.732 ± .007	.729 ± .016	.657 ± .005	.636 ± .005
	LC_{M-T}	.639 ± .004	.575 ± .009	.601 ± .007	.604 ± .008	.727 ± .016	.583 ± .005	.582 ± .006
Slashdot	LC	.215 ± .007	.215 ± .007	.216 ± .007	.346 ± .023	.250 ± .008	.244 ± .041	.225 ± .041
	PC	.217 ± .007	.215 ± .007	.216 ± .006	.219 ± .008	.250 ± .007	—	—
	LC_{S-T}	.276 ± .010	.297 ± .005	.231 ± .008	.216 ± .009	.250 ± .006	.386 ± .020	.248 ± .009
	LC_{M-T}	.294 ± .031	.248 ± .011	.216 ± .007	.217 ± .008	.251 ± .006	.230 ± .008	.229 ± .014
Social	LC	.415 ± .009	.423 ± .006	.414 ± .009	.445 ± .010	.493 ± .013	.411 ± .007	.406 ± .010
	PC	.417 ± .009	.392 ± .008	.406 ± .008	.398 ± .008	.481 ± .010	—	—
	LC_{S-T}	.456 ± .010	.492 ± .004	.391 ± .008	.408 ± .009	.402 ± .006	.508 ± .003	.484 ± .008
	LC_{M-T}	.436 ± .024	.369 ± .007	.388 ± .008	.393 ± .007	.404 ± .008	.391 ± .005	.387 ± .004
Society	LC	.617 ± .007	.647 ± .003	.630 ± .004	.641 ± .009	.722 ± .019	.646 ± .008	.642 ± .007
	PC	.620 ± .005	.569 ± .009	.616 ± .007	.596 ± .008	.720 ± .020	—	—
	LC_{S-T}	.644 ± .008	.624 ± .004	.619 ± .005	.612 ± .005	.589 ± .009	.658 ± .005	.625 ± .006
	LC_{M-T}	.636 ± .026	.582 ± .008	.558 ± .008	.567 ± .008	.587 ± .009	.554 ± .007	.552 ± .008
Yeast	LC	.359 ± .004	.386 ± .005	.366 ± .005	.370 ± .008	.382 ± .002	.376 ± .006	.380 ± .005
	PC	.401 ± .006	.358 ± .007	.374 ± .008	.357 ± .007	.380 ± .003	—	—
	LC_{S-T}	.379 ± .003	.406 ± .034	.368 ± .009	.353 ± .005	.333 ± .004	.463 ± .003	.441 ± .003
	LC_{M-T}	.370 ± .015	.370 ± .009	.329 ± .004	.328 ± .005	.333 ± .004	.333 ± .003	.330 ± .004

TABLE A.5: Ensemble multi-label variant performances in term of *Macro-F1 loss* (Part 1/2). 'LC' denotes the ***Label Combination*** variant, 'LC' denotes the ***Powerset Combination*** variant, 'LC_{M-T}' denotes the Multi-threshold variant, 'LC_{S-T}' denotes the Single-threshold variant.

Data sets	Variants	EBR	ELP	ECC	RFPCT	VPCME	RAkEL	CkMLC
Arts	LC	.531 ± .036	.456 ± .056	.483 ± .038	.582 ± .025	.529 ± .055	.379 ± .043	.549 ± .070
	PC	.596 ± .032	.509 ± .023	.551 ± .030	.630 ± .021	.526 ± .050	—	—
	LC_{S-T}	.513 ± .035	.336 ± .039	.393 ± .026	.599 ± .032	.479 ± .028	.521 ± .031	.514 ± .030
	LC_{M-T}	.483 ± .024	.311 ± .033	.386 ± .036	.495 ± .016	.469 ± .027	.452 ± .051	.347 ± .088
Birds	LC	.374 ± .058	.436 ± .071	.465 ± .086	.480 ± .069	.283 ± .037	.395 ± .034	.458 ± .099
	PC	.510 ± .068	.514 ± .041	.498 ± .053	.530 ± .042	.284 ± .035	—	—
	LC_{S-T}	.341 ± .043	.305 ± .049	.300 ± .047	.518 ± .067	.178 ± .057	.499 ± .045	.524 ± .064
	LC_{M-T}	.283 ± .048	.203 ± .041	.235 ± .064	.388 ± .034	.173 ± .050	.448 ± .104	.382 ± .135
Business	LC	.435 ± .047	.286 ± .052	.438 ± .067	.514 ± .045	.356 ± .033	.379 ± .026	.498 ± .060
	PC	.552 ± .026	.417 ± .049	.455 ± .041	.549 ± .024	.354 ± .037	—	—
	LC_{S-T}	.431 ± .051	.241 ± .032	.249 ± .045	.559 ± .029	.291 ± .040	.482 ± .046	.487 ± .045
	LC_{M-T}	.404 ± .048	.244 ± .028	.247 ± .043	.460 ± .043	.281 ± .039	.311 ± .098	.252 ± .104
Computers	LC	.567 ± .039	.505 ± .045	.520 ± .035	.607 ± .025	.533 ± .035	.610 ± .026	.561 ± .023
	PC	.611 ± .026	.514 ± .020	.587 ± .014	.616 ± .026	.534 ± .035	—	—
	LC_{S-T}	.566 ± .042	.487 ± .044	.516 ± .037	.580 ± .023	.441 ± .050	.510 ± .023	.514 ± .027
	LC_{M-T}	.477 ± .041	.470 ± .044	.496 ± .031	.570 ± .047	.436 ± .037	.507 ± .046	.411 ± .044
Education	LC	.358 ± .020	.235 ± .021	.324 ± .040	.422 ± .025	.342 ± .022	.178 ± .027	.329 ± .011
	PC	.447 ± .049	.291 ± .022	.334 ± .021	.472 ± .056	.342 ± .023	—	—
	LC_{S-T}	.342 ± .021	.202 ± .011	.221 ± .016	.418 ± .060	.267 ± .030	.312 ± .022	.311 ± .029
	LC_{M-T}	.317 ± .040	.220 ± .007	.219 ± .012	.337 ± .026	.267 ± .030	.214 ± .016	.202 ± .010
Emotions	LC	.368 ± .015	.323 ± .014	.320 ± .013	.332 ± .025	.448 ± .088	.520 ± .008	.411 ± .008
	PC	.331 ± .010	.317 ± .013	.332 ± .018	.323 ± .018	.449 ± .088	—	—
	LC_{S-T}	.366 ± .016	.353 ± .032	.347 ± .011	.324 ± .011	.424 ± .086	.387 ± .019	.373 ± .015
	LC_{M-T}	.363 ± .020	.380 ± .016	.363 ± .014	.355 ± .020	.415 ± .077	.437 ± .015	.389 ± .015
Enron	LC	.360 ± .031	.398 ± .034	.381 ± .029	.444 ± .036	.212 ± .026	.444 ± .027	.394 ± .031
	PC	.448 ± .034	.385 ± .032	.417 ± .027	.459 ± .033	.212 ± .026	—	—
	LC_{S-T}	.264 ± .034	.315 ± .032	.389 ± .032	.406 ± .031	.138 ± .023	.355 ± .027	.385 ± .032
	LC_{M-T}	.264 ± .034	.304 ± .036	.323 ± .033	.304 ± .024	.115 ± .021	.217 ± .023	.175 ± .020
Entertainment	LC	.434 ± .018	.403 ± .025	.398 ± .007	.398 ± .013	.498 ± .019	.452 ± .030	.409 ± .004
	PC	.432 ± .026	.388 ± .018	.404 ± .008	.463 ± .042	.499 ± .019	—	—
	LC_{S-T}	.434 ± .018	.406 ± .029	.450 ± .020	.400 ± .018	.444 ± .071	.398 ± .015	.396 ± .013
	LC_{M-T}	.449 ± .019	.398 ± .025	.414 ± .023	.434 ± .018	.440 ± .066	.424 ± .027	.409 ± .024
Flags	LC	.354 ± .022	.360 ± .030	.304 ± .014	.348 ± .014	.325 ± .046	.206 ± .011	.322 ± .011
	PC	.308 ± .018	.316 ± .006	.320 ± .017	.316 ± .021	.302 ± .055	—	—
	LC_{S-T}	.351 ± .022	.336 ± .071	.359 ± .020	.314 ± .014	.347 ± .074	.316 ± .020	.308 ± .016
	LC_{M-T}	.340 ± .042	.321 ± .062	.351 ± .042	.332 ± .017	.323 ± .063	.337 ± .049	.304 ± .054
Health	LC	.400 ± .031	.339 ± .012	.339 ± .024	.349 ± .025	.402 ± .024	.599 ± .024	.491 ± .017
	PC	.356 ± .013	.322 ± .011	.352 ± .028	.332 ± .009	.402 ± .024	—	—
	LC_{S-T}	.395 ± .030	.345 ± .011	.380 ± .003	.331 ± .025	.479 ± .015	.424 ± .021	.407 ± .011
	LC_{M-T}	.395 ± .023	.425 ± .032	.396 ± .019	.381 ± .022	.490 ± .015	.461 ± .012	.429 ± .034

Complementary of Table A.5
Ensemble multi-label variant performances in term of *Macro-F1 loss* (Part 2/2).
'LC' denotes the ***Label Combination*** variant, 'LC' denotes the ***Powerset Combination*** variant,
'LC_{M-T}' denotes the Multi-threshold variant, 'LC_{S-T}' denotes the Single-threshold variant.

Data sets (↓)	Combination	EBR	ELP	ECC	RFPCT	VPCME	RAkEL	CkMLC
Image	LC	.110 ± .009	.104 ± .016	.176 ± .009	.139 ± .012	.129 ± .012	.024 ± .012	.181 ± .007
	PC	.168 ± .011	.138 ± .011	.157 ± .013	.196 ± .013	.123 ± .013	—	—
	LC_{S-T}	.106 ± .008	.079 ± .026	.128 ± .013	.175 ± .009	.112 ± .018	.146 ± .015	.152 ± .013
	LC_{M-T}	.102 ± .008	.093 ± .012	.117 ± .011	.124 ± .009	.110 ± .015	.112 ± .021	.093 ± .010
Medical	LC	.316 ± .021	.230 ± .016	.242 ± .050	.257 ± .024	.278 ± .013	.674 ± .016	.422 ± .016
	PC	.264 ± .016	.231 ± .017	.243 ± .014	.255 ± .015	.275 ± .010	—	—
	LC_{S-T}	.312 ± .015	.249 ± .012	.290 ± .019	.259 ± .022	.308 ± .009	.342 ± .023	.332 ± .025
	LC_{M-T}	.301 ± .018	.335 ± .013	.311 ± .017	.304 ± .028	.317 ± .008	.368 ± .022	.335 ± .021
Recreation	LC	.341 ± .051	.271 ± .023	.281 ± .032	.346 ± .030	.317 ± .049	.382 ± .027	.351 ± .038
	PC	.383 ± .021	.303 ± .026	.321 ± .021	.402 ± .029	.318 ± .049	—	—
	LC_{S-T}	.334 ± .050	.300 ± .028	.303 ± .048	.358 ± .041	.277 ± .053	.296 ± .020	.294 ± .025
	LC_{M-T}	.326 ± .035	.290 ± .035	.289 ± .028	.315 ± .016	.284 ± .050	.285 ± .037	.244 ± .021
Reference	LC	.502 ± .031	.447 ± .022	.520 ± .013	.571 ± .029	.428 ± .011	.552 ± .029	.532 ± .021
	PC	.614 ± .030	.515 ± .017	.538 ± .017	.637 ± .036	.429 ± .011	—	—
	LC_{S-T}	.504 ± .027	.343 ± .028	.415 ± .013	.569 ± .037	.306 ± .022	.511 ± .012	.528 ± .029
	LC_{M-T}	.399 ± .024	.343 ± .028	.389 ± .021	.459 ± .027	.292 ± .022	.344 ± .029	.323 ± .014
Scene	LC	.473 ± .009	.418 ± .007	.409 ± .005	.531 ± .008	.410 ± .007	.532 ± .011	.480 ± .006
	PC	.531 ± .010	.423 ± .008	.459 ± .008	.570 ± .006	.408 ± .006	—	—
	LC_{S-T}	.458 ± .010	.309 ± .005	.351 ± .008	.535 ± .006	.330 ± .011	.497 ± .016	.476 ± .016
	LC_{M-T}	.374 ± .023	.276 ± .006	.323 ± .007	.416 ± .008	.326 ± .011	.364 ± .019	.310 ± .009
Science	LC	.364 ± .022	.212 ± .030	.385 ± .052	.418 ± .039	.301 ± .030	.499 ± .027	.405 ± .021
	PC	.504 ± .025	.296 ± .024	.372 ± .024	.540 ± .038	.301 ± .030	—	—
	LC_{S-T}	.363 ± .024	.137 ± .025	.173 ± .029	.449 ± .037	.269 ± .028	.399 ± .032	.384 ± .041
	LC_{M-T}	.298 ± .029	.131 ± .034	.154 ± .023	.273 ± .023	.259 ± .029	.249 ± .029	.176 ± .029
Slashdot	LC	.159 ± .027	.060 ± .017	.210 ± .037	.182 ± .025	.054 ± .019	.415 ± .038	.351 ± .062
	PC	.304 ± .054	.283 ± .052	.208 ± .041	.433 ± .057	.097 ± .027	—	—
	LC_{S-T}	.161 ± .031	.030 ± .021	.046 ± .020	.366 ± .049	.047 ± .017	.286 ± .024	.309 ± .046
	LC_{M-T}	.157 ± .028	.041 ± .021	.041 ± .012	.155 ± .026	.047 ± .017	.209 ± .061	.156 ± .086
Social	LC	.511 ± .053	.445 ± .027	.462 ± .059	.490 ± .044	.465 ± .037	.486 ± .029	.483 ± .024
	PC	.543 ± .040	.456 ± .015	.504 ± .024	.473 ± .026	.465 ± .037	—	—
	LC_{S-T}	.513 ± .046	.470 ± .031	.477 ± .023	.458 ± .057	.393 ± .035	.445 ± .056	.447 ± .045
	LC_{M-T}	.513 ± .027	.470 ± .028	.501 ± .015	.467 ± .037	.375 ± .030	.437 ± .040	.391 ± .019
Society	LC	.502 ± .048	.447 ± .034	.520 ± .047	.571 ± .027	.428 ± .040	.552 ± .032	.532 ± .041
	PC	.614 ± .025	.515 ± .028	.538 ± .035	.637 ± .034	.429 ± .040	—	—
	LC_{S-T}	.504 ± .041	.343 ± .036	.415 ± .032	.569 ± .022	.306 ± .028	.511 ± .025	.528 ± .016
	LC_{M-T}	.399 ± .053	.343 ± .036	.389 ± .020	.459 ± .021	.292 ± .035	.344 ± .019	.323 ± .022
Yeast	LC	.531 ± .018	.336 ± .004	.487 ± .037	.505 ± .056	.430 ± .045	.548 ± .003	.518 ± .003
	PC	.533 ± .039	.478 ± .006	.490 ± .026	.507 ± .031	.431 ± .045	—	—
	LC_{S-T}	.521 ± .031	.258 ± .078	.315 ± .070	.499 ± .024	.301 ± .058	.505 ± .037	.517 ± .037
	LC_{M-T}	.492 ± .031	.258 ± .078	.292 ± .019	.450 ± .042	.293 ± .053	.378 ± .060	.244 ± .022

TABLE A.6: Ensemble multi-label variant performances in term of *Hamming loss* (Part 1/2). 'LC' denotes the ***Label Combination*** variant, 'LC' denotes the ***Powerset Combination*** variant, 'LC_{M-T}' denotes the Multi-threshold variant, 'LC_{S-T}' denotes the Single-threshold variant.

Data sets	Variants	EBR	ELP	ECC	RFPCT	VPCME	RAkEL	CkMLC
Arts	LC	.055 ± .001	.055 ± .001	.054 ± .001	.054 ± .001	.059 ± .001	.060 ± .018	.064 ± .030
	PC	.059 ± .001	.062 ± .001	.055 ± .001	.066 ± .001	.059 ± .001	—	—
	LC_{S-T}	.063 ± .001	.056 ± .001	.054 ± .001	.054 ± .001	.058 ± .001	.166 ± .022	.075 ± .009
	LC_{M-T}	.058 ± .001	.053 ± .001	.054 ± .001	.054 ± .001	.058 ± .001	.063 ± .026	.056 ± .005
Birds	LC	.048 ± .002	.050 ± .002	.046 ± .002	.048 ± .002	.050 ± .002	.063 ± .022	.065 ± .025
	PC	.049 ± .002	.049 ± .002	.047 ± .002	.055 ± .003	.050 ± .002	—	—
	LC_{S-T}	.048 ± .003	.046 ± .001	.045 ± .002	.049 ± .002	.049 ± .002	.114 ± .021	.069 ± .012
	LC_{M-T}	.048 ± .002	.045 ± .002	.045 ± .002	.048 ± .002	.049 ± .002	.066 ± .028	.053 ± .008
Business	LC	.026 ± .001	.026 ± .001	.026 ± .001	.026 ± .001	.026 ± .001	.028 ± .005	.028 ± .008
	PC	.027 ± .001	.026 ± .001	.026 ± .001	.028 ± .001	.026 ± .001	—	—
	LC_{S-T}	.027 ± .001	.026 ± .001	.026 ± .001	.026 ± .001	.026 ± .001	.057 ± .005	.036 ± .002
	LC_{M-T}	.027 ± .001	.025 ± .001	.025 ± .001	.026 ± .001	.026 ± .001	.029 ± .007	.027 ± .003
Computers	LC	.035 ± .001	.035 ± .001	.034 ± .001	.035 ± .001	.039 ± .001	.036 ± .001	.035 ± .001
	PC	.038 ± .001	.037 ± .001	.035 ± .001	.040 ± .001	.039 ± .001	—	—
	LC_{S-T}	.039 ± .001	.038 ± .001	.034 ± .001	.035 ± .001	.039 ± .001	.088 ± .001	.048 ± .001
	LC_{M-T}	.037 ± .001	.034 ± .001	.033 ± .001	.035 ± .001	.038 ± .001	.036 ± .001	.036 ± .001
Education	LC	.038 ± .001	.038 ± .001	.038 ± .001	.039 ± .001	.039 ± .001	.038 ± .001	.038 ± .001
	PC	.040 ± .001	.044 ± .001	.038 ± .001	.046 ± .001	.039 ± .001	—	—
	LC_{S-T}	.043 ± .001	.041 ± .001	.038 ± .001	.039 ± .001	.038 ± .001	.116 ± .001	.054 ± .001
	LC_{M-T}	.040 ± .001	.037 ± .001	.037 ± .001	.039 ± .001	.038 ± .001	.038 ± .001	.038 ± .001
Emotions	LC	.197 ± .008	.189 ± .006	.187 ± .007	.196 ± .011	.261 ± .005	.238 ± .009	.199 ± .008
	PC	.197 ± .009	.198 ± .007	.189 ± .007	.208 ± .014	.261 ± .006	—	—
	LC_{S-T}	.203 ± .013	.187 ± .008	.190 ± .012	.193 ± .011	.255 ± .009	.368 ± .037	.301 ± .006
	LC_{M-T}	.196 ± .010	.186 ± .005	.191 ± .009	.195 ± .011	.255 ± .010	.238 ± .009	.234 ± .010
Enron	LC	.046 ± .001	.048 ± .001	.046 ± .001	.049 ± .001	.049 ± .001	.047 ± .001	.047 ± .001
	PC	.052 ± .001	.057 ± .001	.052 ± .001	.064 ± .001	.049 ± .001	—	—
	LC_{S-T}	.048 ± .001	.048 ± .001	.046 ± .001	.049 ± .001	.049 ± .001	.107 ± .003	.058 ± .001
	LC_{M-T}	.046 ± .001	.047 ± .001	.045 ± .001	.049 ± .001	.049 ± .001	.048 ± .001	.046 ± .001
Entertainment	LC	.054 ± .001	.053 ± .001	.051 ± .001	.053 ± .001	.061 ± .001	.053 ± .001	.052 ± .001
	PC	.057 ± .001	.060 ± .001	.052 ± .001	.063 ± .001	.061 ± .001	—	—
	LC_{S-T}	.057 ± .001	.054 ± .001	.051 ± .001	.053 ± .001	.060 ± .001	.158 ± .003	.074 ± .001
	LC_{M-T}	.056 ± .001	.050 ± .001	.050 ± .001	.053 ± .001	.059 ± .001	.053 ± .001	.053 ± .001
Flags	LC	.261 ± .010	.247 ± .010	.248 ± .010	.260 ± .010	.325 ± .013	.271 ± .017	.251 ± .015
	PC	.267 ± .010	.271 ± .013	.262 ± .011	.273 ± .006	.329 ± .011	—	—
	LC_{S-T}	.263 ± .012	.249 ± .011	.248 ± .009	.265 ± .011	.330 ± .009	.391 ± .050	.318 ± .026
	LC_{M-T}	.262 ± .011	.257 ± .011	.251 ± .016	.260 ± .010	.334 ± .012	.266 ± .020	.270 ± .016
Health	LC	.035 ± .001	.035 ± .001	.033 ± .001	.034 ± .001	.041 ± .001	.034 ± .001	.033 ± .001
	PC	.036 ± .001	.034 ± .001	.034 ± .001	.036 ± .001	.041 ± .002	—	—
	LC_{S-T}	.036 ± .001	.034 ± .001	.033 ± .001	.034 ± .001	.039 ± .002	.088 ± .001	.044 ± .001
	LC_{M-T}	.036 ± .001	.031 ± .001	.032 ± .001	.034 ± .001	.039 ± .002	.034 ± .001	.034 ± .001

Complementary of Table A.6
Ensemble multi-label variant performances in term of *Hamming loss* (Part 2/2).
'LC' denotes the ***Label Combination*** variant, 'LC' denotes the ***Powerset Combination*** variant, 'LC_{M-T}' denotes the Multi-threshold variant, 'LC_{S-T}' denotes the Single-threshold variant.

Data sets (↓)	Combination	EBR	ELP	ECC	RFPCT	VPCME	RAkEL	CkMLC
Image	LC	.164 ± .004	.158 ± .003	.154 ± .003	.155 ± .003	.171 ± .002	.196 ± .007	.160 ± .003
	PC	.165 ± .004	.157 ± .007	.156 ± .005	.165 ± .005	.170 ± .003	—	—
	LC_{S-T}	.172 ± .006	.151 ± .004	.154 ± .003	.156 ± .003	.172 ± .004	.380 ± .037	.273 ± .007
	LC_{M-T}	.161 ± .004	.151 ± .003	.154 ± .003	.155 ± .003	.173 ± .004	.197 ± .010	.186 ± .005
Medical	LC	.011 ± .001	.015 ± .001	.017 ± .001	.011 ± .001	.013 ± .001	.010 ± .001	.010 ± .001
	PC	.010 ± .001	.014 ± .001	.017 ± .001	.012 ± .001	.013 ± .001	—	—
	LC_{S-T}	.010 ± .001	.014 ± .001	.013 ± .001	.012 ± .001	.013 ± .001	.016 ± .001	.014 ± .002
	LC_{M-T}	.010 ± .001	.011 ± .001	.011 ± .001	.011 ± .001	.013 ± .001	.010 ± .001	.010 ± .001
Recreation	LC	.055 ± .001	.055 ± .001	.054 ± .001	.054 ± .001	.060 ± .001	.054 ± .001	.053 ± .001
	PC	.060 ± .001	.063 ± .001	.054 ± .001	.068 ± .001	.060 ± .001	—	—
	LC_{S-T}	.063 ± .002	.055 ± .001	.054 ± .001	.054 ± .001	.059 ± .001	.171 ± .002	.072 ± .001
	LC_{M-T}	.059 ± .001	.053 ± .001	.053 ± .001	.054 ± .001	.058 ± .001	.054 ± .001	.054 ± .001
Reference	LC	.026 ± .001	.026 ± .001	.025 ± .001	.026 ± .001	.030 ± .001	.026 ± .001	.026 ± .001
	PC	.028 ± .001	.029 ± .001	.026 ± .001	.030 ± .001	.030 ± .001	—	—
	LC_{S-T}	.030 ± .001	.028 ± .001	.025 ± .001	.026 ± .001	.030 ± .001	.064 ± .001	.033 ± .001
	LC_{M-T}	.028 ± .001	.025 ± .001	.025 ± .001	.026 ± .001	.029 ± .001	.026 ± .001	.026 ± .001
Scene	LC	.091 ± .002	.093 ± .001	.089 ± .001	.091 ± .002	.089 ± .002	.117 ± .007	.095 ± .002
	PC	.094 ± .002	.082 ± .002	.087 ± .002	.092 ± .002	.088 ± .002	—	—
	LC_{S-T}	.091 ± .003	.082 ± .001	.081 ± .001	.091 ± .003	.088 ± .003	.267 ± .044	.147 ± .003
	LC_{M-T}	.093 ± .002	.082 ± .001	.082 ± .002	.092 ± .002	.089 ± .003	.117 ± .007	.114 ± .005
Science	LC	.033 ± .001	.033 ± .001	.032 ± .001	.033 ± .001	.034 ± .001	.032 ± .001	.032 ± .001
	PC	.036 ± .001	.039 ± .001	.033 ± .001	.041 ± .001	.034 ± .001	—	—
	LC_{S-T}	.038 ± .001	.033 ± .001	.032 ± .001	.033 ± .001	.033 ± .001	.085 ± .001	.040 ± .001
	LC_{M-T}	.035 ± .001	.031 ± .001	.032 ± .001	.033 ± .001	.033 ± .001	.032 ± .001	.032 ± .001
Slashdot	LC	.015 ± .001	.015 ± .001	.015 ± .001	.015 ± .001	.019 ± .001	.017 ± .002	.017 ± .005
	PC	.015 ± .001	.015 ± .001	.015 ± .001	.016 ± .001	.019 ± .001	—	—
	LC_{S-T}	.015 ± .001	.015 ± .001	.015 ± .001	.016 ± .001	.019 ± .001	.039 ± .005	.020 ± .001
	LC_{M-T}	.015 ± .001	.015 ± .001	.015 ± .001	.016 ± .001	.019 ± .001	.018 ± .006	.016 ± .001
Social	LC	.021 ± .001	.020 ± .001	.020 ± .001	.021 ± .001	.022 ± .001	.021 ± .001	.020 ± .001
	PC	.023 ± .001	.023 ± .001	.020 ± .001	.024 ± .001	.022 ± .001	—	—
	LC_{S-T}	.023 ± .001	.021 ± .001	.020 ± .001	.021 ± .001	.022 ± .001	.050 ± .001	.027 ± .001
	LC_{M-T}	.023 ± .001	.020 ± .001	.019 ± .001	.021 ± .001	.022 ± .001	.021 ± .001	.021 ± .001
Society	LC	.052 ± .001	.053 ± .001	.052 ± .001	.054 ± .001	.055 ± .001	.053 ± .001	.052 ± .001
	PC	.058 ± .001	.057 ± .001	.053 ± .001	.065 ± .002	.055 ± .001	—	—
	LC_{S-T}	.058 ± .001	.057 ± .002	.052 ± .001	.054 ± .001	.055 ± .001	.156 ± .004	.065 ± .001
	LC_{M-T}	.057 ± .001	.052 ± .001	.052 ± .001	.054 ± .001	.054 ± .001	.053 ± .001	.052 ± .001
Yeast	LC	.195 ± .002	.198 ± .002	.193 ± .002	.195 ± .003	.196 ± .001	.197 ± .003	.197 ± .002
	PC	.216 ± .003	.206 ± .004	.205 ± .003	.210 ± .004	.196 ± .001	—	—
	LC_{S-T}	.203 ± .005	.198 ± .003	.193 ± .002	.195 ± .002	.195 ± .003	.509 ± .009	.302 ± .003
	LC_{M-T}	.195 ± .002	.194 ± .002	.193 ± .003	.195 ± .003	.195 ± .003	.197 ± .003	.195 ± .002

Bibliography

[1] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

[2] Eva Gibaja and Sebastián Ventura. A tutorial on multilabel learning. *ACM Computing Surveys (CSUR)*, 47(3):52, 2015.

[3] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2010.

[4] Eleftherios Spyromitros, Grigorios Tsoumakas, and Ioannis Vlahavas. An empirical study of lazy multilabel classification algorithms. In *Hellenic conference on Artificial Intelligence*, pages 401–406. Springer, 2008.

[5] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

[6] Amanda Clare and Ross D King. Knowledge discovery in multi-label phenotype data. In *Principles of data mining and knowledge discovery*, pages 42–53. Springer, 2001.

[7] Hendrik Blockeel, Luc De Raedt, and Jan Ramon. Top-down induction of clustering trees. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 55–63, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8. URL http://dl.acm.org/citation.cfm?id=645527.657456.

[8] Min-Ling Zhang and Zhi-Hua Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.

[9] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 2012.

[10] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[11] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[12] Cha Zhang and Yunqian Ma. *Ensemble machine learning*, volume 1. Springer, 2012.

[13] Robert E Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

[14] Fazia Bellal, Haytham Elghazel, and Alex Aussem. A semi-supervised feature ranking method with ensemble learning. *Pattern Recognition Letters*, 33(10):1426–1433, 2012.

[15] G. Tsoumakas, I. Katakis, and I. Vlahavas. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 2011.

[16] Jesse Read, Bernhard Pfahringer, and Geoff Holmes. Multi-label classification using ensembles of pruned sets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 995–1000. IEEE, 2008.

[17] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359, 2011.

[18] Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1-2):5–45, 2012.

[19] Jesse Read. A pruned problem transformation method for multi-label classification. In *Proc. 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008)*, volume 143150, 2008.

[20] Maxime Gasse, Alex Aussem, and Haytham Elghazel. On the optimality of multi-label classification under subset zero-one loss for distributions satisfying the composition property. In *International Conference on Machine Learning*, volume 37, pages 2531–2539, 2015.

[21] Maxime Gasse, Alex Aussem, and Haytham Elghazel. A hybrid algorithm for bayesian network structure learning with application to multi-label learning. *Expert Systems with Applications*, 41(15):6755–6772, 2014.

[22] Yiming Yang. A study of thresholding strategies for text categorization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 137–145. ACM, 2001.

[23] Marios I., George S., G. Tsoumakas, and I. Vlahavas. Obtaining bipartitions from score vectors for multi-label classification. In *ICTAI*, 2010.

[24] I. Pillai, G. Fumera, and F. Roli. Threshold optimisation for multi-label classifiers. *Pattern Recognition*, 2013.

[25] R. Fan and C. Lin. A study on threshold selection for multi-label classification. *Department of Computer Science, National Taiwan University*, 2007.

[26] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3), 2006.

[27] Eva Gibaja and Sebastián Ventura. Multi-label learning: a review of the state of the art and ongoing research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(6):411–444, 2014.

[28] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2014.

[29] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *NIPS*, volume 14, pages 681–687, 2001.

[30] Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185, 2008.

[31] Hendrik Blockeel, Leander Schietgat, Jan Struyf, Sašo Džeroski, and Amanda Clare. Decision trees for hierarchical multilabel classification: A case study in functional genomics. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 18–29. Springer, 2006.

[32] Dragi Kocev, Celine Vens, Jan Struyf, and Sašo Džeroski. *Ensembles of multi-objective decision trees*. Springer, 2007.

[33] Yonatan Amit, Ofer Dekel, and Yoram Singer. A boosting algorithm for label covering in multilabel problems. In *AISTATS*, pages 27–34, 2007.

[34] Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, pages 377–384. ACM, 2005.

[35] James Petterson and Tibério S Caetano. Submodular multi-label learning. In *Advances in Neural Information Processing Systems*, pages 1512–1520, 2011.

[36] Thomas Finley and Thorsten Joachims. Training structural svms when exact inference is intractable. In *Proceedings of the 25th international conference on Machine learning*, pages 304–311. ACM, 2008.

[37] Bharath Hariharan, Lihi Zelnik-Manor, Manik Varma, and Svn Vishwanathan. Large scale max-margin multi-label classification with priors. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 423–430, 2010.

[38] Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30. Springer, 2004.

[39] Patrick Pletscher, Cheng Soon Ong, and Joachim M Buhmann. Entropy and margin maximization for structured output learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 83–98. Springer, 2010.

[40] Andrew McCallum. Multi-label text classification with a mixture model trained by em. In *AAAI'99 workshop on text learning*, pages 1–7, 1999.

[41] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[42] Martin D Buhmann. *Radial basis functions: theory and implementations*, volume 12. Cambridge university press, 2003.

[43] Min-Ling Zhang. Ml-rbf: Rbf neural networks for multi-label learning. *Neural Processing Letters*, 29(2):61–74, 2009.

[44] Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16):1897–1916, 2008.

[45] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine learning*, 73(2):133–153, 2008.

[46] Klaus Brinker, Johannes Fürnkranz, and Eyke Hüllermeier. A unified model for multilabel classification and ranking. In *Proceedings of the 2006 conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29–September 1, 2006, Riva del Garda, Italy*, pages 489–493. IOS Press, 2006.

[47] Eneldo Loza Mencía, Sang-Hyeun Park, and Johannes Fürnkranz. Efficient voting prediction for pairwise multilabel classification. *Neurocomputing*, 73(7):1164–1176, 2010.

[48] Gjorgji Madjarov, Dejan Gjorgjevikj, and Sašo Džeroski. Dual layer voting method for efficient multi-label classification. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 232–239. Springer, 2011.

[49] Weiwei Cheng, Eyke Hüllermeier, and Krzysztof J Dembczynski. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 279–286, 2010.

[50] Krzysztof Dembczyński, Willem Waegeman, and Eyke Hüllermeier. An analysis of chaining in multi-label classification. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 294–299. IOS Press, 2012.

[51] Deiner Mena, Elena Montañés, José R Quevedo, and Juan José Del Coz. Using a* for inference in probabilistic classifier chains. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 3707–3713. AAAI Press, 2015.

[52] Abhishek Kumar, Shankar Vembu, Aditya Krishna Menon, and Charles Elkan. Beam search algorithms for multilabel learning. *Machine learning*, 92(1):65–89, 2013.

[53] Everton Alvares-Cherman, Jean Metz, and Maria Carolina Monard. Incorporating label dependency into the binary relevance framework for multi-label classification. *Expert Systems with Applications*, 39(2):1647–1655, 2012.

[54] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, pages 30–44, 2008.

[55] Dragi Kocev, Celine Vens, Jan Struyf, and Sašo Džeroski. Tree ensembles for predicting structured outputs. *Pattern Recognition*, 46(3):817–833, 2013.

[56] Krzysztof Dembszynski, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. On label dependence in multilabel classification. In *LastCFP: ICML Workshop on Learning from Multi-label data*. Ghent University, KERMIT, Department of Applied Mathematics, Biometrics and Process Control, 2010.

[57] Nadia Ghamrawi and Andrew McCallum. Collective multi-label classification. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 195–200. ACM, 2005.

[58] Nagarajan Natarajan, Oluwasanmi Koyejo, Pradeep Ravikumar, and Inderjit Dhillon. Optimal classification with multivariate losses. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1530–1538, 2016.

[59] David D Lewis. Evaluating and optimizing autonomous text classification systems. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 246–254. ACM, 1995.

[60] Krzysztof J Dembczynski, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. An exact algorithm for f-measure maximization. In *Advances in neural information processing systems*, pages 1404–1412, 2011.

[61] Flavio Chierichetti, Ravi Kumar, Sandeep Pandey, and Sergei Vassilvitskii. Finding the jaccard median. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 293–311. SIAM, 2010.

[62] Maxime Gasse and Alex Aussem. F-measure maximization in multi-label classification with conditionally independent label subsets. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 619–631. Springer, 2016.

[63] Krzysztof Dembczynski, Arkadiusz Jachnik, Wojciech Kotlowski, Willem Waegeman, and Eyke Hüllermeier. Optimizing the f-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. *ICML (3)*, 28:1130–1138, 2013.

[64] Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. Regret analysis for performance metrics in multi-label classification: the case of hamming and subset zero-one loss. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 280–295. Springer, 2010.

[65] Lei Tang, Suju Rajan, and Vijay K Narayanan. Large scale multi-label classification via metalabeler. In *Proceedings of the 18th international conference on World wide web*, pages 211–220. ACM, 2009.

[66] José Ramón Quevedo, Oscar Luaces, and Antonio Bahamonde. Multilabel classifiers with a probabilistic thresholding strategy. *Pattern Recognition*, 45(2):876–883, 2012.

[67] Zachary C Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize f1 measure. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–239. Springer, 2014.

[68] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

[69] Robert E Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336, 1999.

[70] Yoav Freund, Yishay Mansour, and Robert E Schapire. Generalization bounds for averaged classifiers. *Annals of Statistics*, pages 1698–1722, 2004.

[71] Padhraic Smyth and David Wolpert. Linearly combining density estimators via stacking. *Machine Learning*, 36(1-2):59–83, 1999.

[72] Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.

[73] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[74] L Breiman. Out-of-bag estimation. *Technical report, Statistics Department, University of California, Berkeley*, 199, 1996.

[75] Robert Tibshirani. *Bias, variance and prediction error for classification rules*. University of Toronto, Department of Statistics, 1996.

[76] David H Wolpert and William G Macready. An efficient method to estimate bagging's generalization error. *Machine Learning*, 35(1):41–55, 1999.

[77] Ping Li, Hong Li, and Min Wu. Multi-label ensemble based on variable pairwise constraint projection. *Information Sciences*, 222:269–281, 2013.

[78] Hendrik Blockeel, Luc De Raedt, and Jan Ramon. Top-down induction of clustering trees. *arXiv preprint cs/0011032*, 2000.

[79] Gulisong Nasierding, Abbas Z Kouzani, and Grigorios Tsoumakas. A triple-random ensemble classification method for mining multi-label data. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 49–56. IEEE, 2010.

[80] L. Rokach, A. Schclar, and E. Itach. Ensemble methods for multi-label classification. *Expert Systems with Applications*, 2014.

[81] Ouadie Gharroudi, Haytham Elghazel, and Alex Aussem. Calibrated k-labelsets for ensemble multi-label classification. In *International Conference on Neural Information Processing*, pages 573–582. Springer, 2015.

[82] Sang-Hyeun Park and Johannes Fürnkranz. Efficient pairwise classification. In *Machine Learning: ECML 2007*, pages 658–665. Springer, 2007.

[83] Robert E Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2-3):135–168, 2000.

[84] Francesco De Comité, Rémi Gilleron, and Marc Tommasi. Learning multi-label alternating decision trees from texts and data. In *Machine Learning and Data Mining in Pattern Recognition*, pages 35–49. Springer, 2003.

[85] Fabrizio Sebastiani, Alessandro Sperduti, and Nicola Valdambrini. An improved boosting algorithm and its application to text categorization. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 78–85. ACM, 2000.

[86] Muhammad Atif Tahir, Josef Kittler, and Ahmed Bouridane. Multilabel classification using heterogeneous ensemble of multi-label classifiers. *Pattern Recognition Letters*, 33 (5):513–523, 2012.

[87] Grigorios Tsoumakas, Eleftherios Spyromitros Xioufis, Jozef Vilcek, and Ioannis P. Vlahavas. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414, 2011.

[88] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. In *Encyclopedia of database systems*, pages 532–538. Springer, 2009.

[89] Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[90] Giovanni Seni and John F Elder. Ensemble methods in data mining: improving accuracy through combining predictions. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1):1–126, 2010.

[91] Gavin Brown. Ensemble learning. In *Encyclopedia of Machine Learning*, pages 312–320. Springer, 2011.

[92] Amanda Clare and Ross D. King. Knowledge discovery in multi-label phenotype data. In *PKDD*, pages 42–53, 2001.

[93] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[94] Min-Ling Zhang and Zhi-Hua Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.

[95] Huan Liu and Hiroshi Motoda. *Computational methods of feature selection*. CRC Press, 2007.

[96] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[97] Sareewan Dendamrongvit, Peerapon Vateekul, and Miroslav Kubat. Irrelevant attributes and imbalanced classes in multi-label text-categorization domains. *Intelligent Data Analysis*, 15(6):843–859, 2011.

[98] Newton SpolaôR, Everton Alvares Cherman, Maria Carolina Monard, and Huei Diana Lee. A comparison of multi-label feature selection methods using the problem transformation approach. *Electronic Notes in Theoretical Computer Science*, 292: 135–151, 2013.

[99] Zheng Zhao, Fred Morstatter, Shashvata Sharma, Salem Alelyani, Aneeth Anand, and Huan Liu. Advancing feature selection research. *ASU feature selection repository*, pages 1–28, 2010.

[100] Konstantinos Trohidis, Grigorios Tsoumakas, George Kalliris, and Ioannis P Vlahavas. Multi-label classification of music into emotions. In *ISMIR*, volume 8, pages 325–330, 2008.

[101] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[102] Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering*, 17(4):491–502, 2005.

[103] Newton Spolaôr, Maria Carolina Monard, Grigorios Tsoumakas, and Huei Diana Lee. A systematic review of multi-label feature selection and a new method based on label construction. *Neurocomputing*, 180:3–15, 2016.

[104] Newton Spolaôr, Everton Alvares Cherman, Maria Carolina Monard, and Huei Diana Lee. A comparison of multi-label feature selection methods using the problem transformation approach. *Electr. Notes Theor. Comput. Sci.*, 292:135–151, 2013.

[105] Weizhu Chen, Jun Yan, Benyu Zhang, Zheng Chen, and Qiang Yang. Document transformation for multi-label feature selection in text categorization. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 451–456. IEEE, 2007.

[106] Qu Wei, Zhang Yang, Zhu Junping, and Yong Wang. Semi-supervised multi-label learning algorithm using dependency among labels. In *International Conference on Machine Learning and Computing*, pages 112–116, 2009.

[107] Newton Spolaôr, Everton Alvares Cherman, Maria Carolina Monard, and Huei Diana Lee. Filter approach feature selection methods to support multi-label learning based on relieff and information gain. In *Advances in Artificial Intelligence-SBIA 2012*, pages 72–81. Springer, 2012.

[108] Min-Ling Zhang, José M Peña, and Victor Robles. Feature selection for multi-label naive bayes classification. *Information Sciences*, 179(19):3218–3229, 2009.

[109] Gauthier Doquire and Michel Verleysen. Mutual information-based feature selection for multilabel classification. *Neurocomputing*, 122:148–155, 2013.

[110] Gerardo Lastra, Oscar Luaces, Jose Quevedo, and Antonio Bahamonde. Graphical feature selection for multilabel classification tasks. *Advances in Intelligent Data Analysis X*, pages 246–257, 2011.

[111] Jae-Sung Lee and Dae-Won Kim. Feature selection for multi-label classification using multivariate mutual information. *Pattern Recognition Letters*, 34(3):349–357, 2013.

[112] Huan Shao, GuoZheng Li, GuoPing Liu, and YiQin Wang. Symptom selection for multi-label data of inquiry diagnosis in traditional chinese medicine. *Science China Information Sciences*, 56(5):1–13, 2013.

[113] Andrea Esuli, Tiziano Fagni, and Fabrizio Sebastiani. Boosting multi-label hierarchical text categorization. *Information Retrieval*, 11(4):287–313, 2008.

[114] Quanquan Gu, Zhenhui Li, and Jiawei Han. Correlated multi-label feature selection. In *CIKM*, pages 1087–1096, 2011.

[115] Yi Liu, Rong Jin, and Liu Yang. Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *Proceedings of the national conference on artificial intelligence*, page 421. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[116] Gang Chen, Yangqiu Song, Fei Wang, and Changshui Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *SDM*, pages 410–419. SIAM, 2008.

[117] Jingdong Wang, Yinghai Zhao, Xiuqing Wu, and Xian-Sheng Hua. A transductive multi-label learning approach for video concept detection. *Pattern Recognition*, 44(10): 2274–2286, 2011.

[118] Yuhong Guo and Dale Schuurmans. Semi-supervised multi-label classification. In *Machine Learning and Knowledge Discovery in Databases*, pages 355–370. Springer, 2012.

[119] Zheng-Jun Zha, Tao Mei, Jingdong Wang, Zengfu Wang, and Xian-Sheng Hua. Graph-based semi-supervised learning with multiple labels. *Journal of Visual Communication and Image Representation*, 20(2):97–103, 2009.

[120] Le Wu and Min-Ling Zhang. Multi-label classification with unlabeled data: An inductive approach. In *Asian Conference on Machine Learning*, pages 197–212, 2013.

[121] Olivier Chapelle, Vikas Sindhwani, and Sathiya S Keerthi. Optimization techniques for semi-supervised support vector machines. *The Journal of Machine Learning Research*, 9:203–233, 2008.

[122] Wang Zhan and Min-Ling Zhang. Inductive semi-supervised multi-label learning with co-training. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1305–1314. ACM, 2017.

[123] Xiaojun Chang, Feiping Nie, Yi Yang, and Heng Huang. A convex formulation for semi-supervised multi-label feature selection. In *AAAI*, pages 1171–1177, 2014.

[124] Abdelouahid Alalga, Khalid Benabdeslem, and Nora Taleb. Soft-constrained laplacian score for semi-supervised multi-label feature selection. *Knowledge and Information Systems*, 47(1):75–98, 2016.

[125] Xiao-dong Wang, Rung-Ching Chen, Chao-qun Hong, Zhi-qiang Zeng, and Zhi-li Zhou. Semi-supervised multi-label feature selection via label correlation analysis with l 1-norm graph embedding. *Image and Vision Computing*, 2017.

[126] Buyue Qian and Ian Davidson. Semi-supervised dimension reduction for multi-label classification. In *AAAI*, volume 10, pages 569–574, 201.

[127] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[128] Hasna Barkia, Haytham Elghazel, and Alex Aussem. Semi-supervised feature importance evaluation with ensemble learning. In *ICDM*, pages 31–40, 2011.

[129] Yi Hong, Sam Kwong, Yuchou Chang, and Qingsheng Ren. Consensus unsupervised feature ranking from multiple views. *Pattern Recognition Letters*, 29(5):595–602, 2008.

[130] Yi Hong, Sam Kwong, Yuchou Chang, and Qingsheng Ren. Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. *Pattern Recognition*, 41(9):2742–2756, 2008.

[131] Haytham Elghazel and Alex Aussem. Unsupervised feature selection with ensemble learning. *Machine Learning*, pages 1–24, 2013.

[132] Ouadie Gharroudi, Haytham Elghazel, and Alex Aussem. A comparison of multi-label feature selection methods using the random forest paradigm. In *Canadian Conference on Artificial Intelligence*, pages 95–106. Springer, 2014.

[133] Krzysztof Dembczynski, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1-2):5–45, 2012.

[134] Dragi Kocev, Celine Vens, Jan Struyf, and Saso Dzeroski. Tree ensembles for predicting structured outputs. *Pattern Recognition*, 46(3):817–833, 2013.

[135] Hendrik Blockeel, Luc De Raedt, and Jan Ramon. Top-down induction of clustering trees. In *ICML*, pages 55–63, 1998.

[136] Dragi Kocev, Ivica Slavkov, and Saso Dzeroski. More is better: Ranking with multiple targets for biomarker discovery. In *2nd International Workshop on Machine Learning in Systems Biology*, page 133, 2008.

[137] Dragi Kocev, Ivica Slavkov, and Saso Dzeroski. Feature ranking for multi-label classification using predictive clustering trees. In *International Workshop on Solving Complex Machine Learning Problems with Ensemble Methods, in conjunction with ECML/PKDD*, pages 56–68, 2013.

[138] Celine Vens and Fabrizio Costa. Random forest based feature induction. In *ICDM*, pages 744–753, 2011.

[139] Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Saso Dzeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45 (9):3084–3104, 2012.

[140] Yvan Saeys, Thomas Abeel, and Yves Van de Peer. Robust feature selection using ensemble feature selection techniques. In *ECML/PKDD (2)*, pages 313–325, 2008.

[141] Ludmila I. Kuncheva. A stability index for feature selection. In *Artificial Intelligence and Applications*, pages 421–427, 2007.

[142] Ouadie Gharroudi, Haytham Elghazel, and Alexandre Aussem. A semi-supervised ensemble approach for multi-label learning. In *IEEE International Conference on Data Mining Workshops, ICDM Workshops*, pages 1197–1204, 2016.

[143] Jiao Wang, Si-wei Luo, and Xian-hua Zeng. A random subspace method for co-training. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 195–200. IEEE, 2008.

[144] Mohamed Farouk Abdel Hady and Friedhelm Schwenker. Combining committee-based semi-supervised learning and active learning. *Journal of Computer Science and Technology*, 25(4):681–698, 2010.

[145] Xiangnan Kong, Michael K Ng, and Zhi-Hua Zhou. Transductive multilabel learning via label set propagation. *Knowledge and Data Engineering, IEEE Transactions on*, 25(3): 704–719, 2013.

[146] Haytham Elghazel, Alex Aussem, Ouadie Gharroudi, and Wafa Saadaoui. Ensemble multi-label text categorization based on rotation forest and latent semantic indexing. *Expert Systems with Applications*, 57:1–11, 2016.

[147] Juan José Rodríguez, Ludmila I. Kuncheva, and Carlos J. Alonso. Rotation forest: A new classifier ensemble method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10): 1619–1630, 2006.

[148] Ludmila I. Kuncheva and Juan José Rodríguez. An experimental study on rotation forest ensembles. In *7th International Workshop of Multiple Classifier Systems (MCS)*, pages 459–468, 2007.

[149] Mohamed Bibimoune, Haytham Elghazel, and Alexandre Aussem. An empirical comparison of supervised ensemble learning approaches. In *International Workshop on Complex Machine Learning Problems with Ensemble Methods COPEM@ECML/PKDD'13*, pages 123–138, 2013.

[150] Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Aikaterini Vrekou, and Ioannis Vlahavas. Multi-target regression via random linear target combinations. In *European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–240. Springer, 2014.

[151] Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233, 2015.

[152] Eleftherios Spyromitros Xioufis, Grigorios Tsoumakas, William Groves, and Ioannis P. Vlahavas. Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*, 104(1):55–98, 2016.