Lucas Henrique Sousa Mello

Analysis of the impacts of label dependence in multi-label learning

Vitória, ES

2021

Lucas Henrique Sousa Mello

Analysis of the impacts of label dependence in multi-label learning

Tese de Doutorado submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Programa de Pós-Graduação em Informática

Supervisor: Prof. Dr. Flávio Miguel Varejão

Co-supervisor: Prof. Dr. Alexandre Loureiros Rodrigues

Vitória, ES

2021

Ficha catalográfica disponibilizada pelo Sistema Integrado de
Bibliotecas - SIBI/UFES e elaborada pelo autor

Sousa Mello, Lucas Henrique, 1991-
S725a Analysis of the impacts of label dependence in multi-label learning / Lucas Henrique Sousa Mello. - 2021.
107 f. : il.

Orientador: Flávio Miguel Varejão.
Coorientador: Alexandre Loureiros Rodrigues.
Tese (Doutorado em Informática) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Aprendizado do computador. 2. Inteligência artificial. 3. Otimização combinatória. I. Miguel Varejão, Flávio. II. Loureiros Rodrigues, Alexandre. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 004

ANALYSIS OF THE IMPACTS OF LABEL DEPENDENCE IN MULTI-LABEL LEARNING

Lucas Henrique Sousa Mello

Tese submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Aprovada em 19 de outubro de 2021:

Prof. Dr. Flávio Miguel Varejão
Orientador

Prof. Dr. Alexandre Loureiros Rodrigues
Coorientador

Prof. Dr. Thiago Oliveira dos Santos
Membro Interno

Prof. Dr. Francisco de Assis Boldt
Membro Externo

Prof. Dr. Edward Hermann Haeusler
Membro Externo

Prof. Dr. Thomas Walter Rauber
Membro Interno

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
Vitória-ES, 19 de outubro de 2021.

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA

O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
FLAVIO MIGUEL VAREJAO - SIAPE 297887
Departamento de Informática - DI/CT
Em 19/10/2021 às 19:36

Para verificar as assinaturas e visualizar o documento original acesse o link:
https://api.lepisma.ufes.br/arquivos-assinados/291981?tipoArquivo=O

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA

O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
THIAGO OLIVEIRA DOS SANTOS - SIAPE 2023810
Departamento de Informática - DI/CT
Em 20/10/2021 às 14:02

Para verificar as assinaturas e visualizar o documento original acesse o link:
https://api.lepisma.ufes.br/arquivos-assinados/292660?tipoArquivo=O

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA

O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
THOMAS WALTER RAUBER - SIAPE 2201072
Departamento de Informática - DI/CT
Em 21/10/2021 às 05:45

Para verificar as assinaturas e visualizar o documento original acesse o link:
https://api.lepisma.ufes.br/arquivos-assinados/293340?tipoArquivo=O

Abstract

Conclusions in the field of multi-label learning are often drawn from experiments using real benchmark datasets, which is a good practice for comparing results. However, it hardly proves or clearly shows how dependencies among class labels impact on the performance and behaviour of multi-label algorithms. A reasonable approach to tackle this issue consists of adopting a mathematical or statistical formulation of the problem and using it to elaborate theoretical proofs. Another approach consists of elaborating experiments in a well-controlled environment where the dependence among labels can be easier controlled and analyzed, which is the case for many works based on artificial datasets. Both approaches are adopted in this thesis to understand the role of label dependence in multi-label learning.

The work done in this thesis is composed of several contributions regarding the analysis of multi-label algorithms from a statistical perspective. One contribution is that calibrated label ranking is an algorithm that can perform extremely poor in particular scenarios where label dependence is present, due to the way that pairwise comparison of labels is done by the algorithm. Another contribution is that the label dependence present in multi-label learning makes the optimization of the expected coverage a NP-HARD problem, even at restricted conditions. Finally, a proposal is presented on how to build an experimental environment where the label dependence can conveniently be controlled for comparing performance among multi-label learning algorithms.

Resumo

Conclusões em aprendizado multirrótulo geralmente são tiradas através de experimentos usando conjuntos de dados reais de referência, o que é uma boa prática ao comparar resultados. No entanto, dificilmente demonstra ou mostra claramente como a dependência entre rótulos afeta o desempenho e o comportamento de algoritmos multirrótulo. Uma abordagem razoável para resolver tal problema consiste em adotar uma formulação matemática ou estatística do problema e usá-lo para elaborar provas teóricas. Outra abordagem consiste em elaborar experimentos em um ambiente controlado, onde a dependência entre rótulos pode ser mais facilmente controlada e analisada, o que é o caso de muitos trabalhos baseados em conjuntos de dados artificiais. Ambas abordagens são adotadas nesta tese para entender o papel da dependência de rótulos na aprendizagem multirrótulo. O trabalho realizado nesta tese é composto de várias contribuições à análise de algoritmos multirrótulo em uma perspectiva estatística. Uma contribuição é que o método *calibrated label ranking* é um algoritmo que pode ter um desempenho extremamente baixo quando empregado em um cenário muito particular em que a dependência entre rótulos está presente, devido à maneira como a comparação em pares de rótulos é feita pelo algoritmo. Outra contribuição é que a dependência entre rótulos a otimização de *coverage* esperado é um problema NP-DIFÍCIL. Por final, é apresentada uma forma de criar um ambiente experimental em que a dependência entre rótulos possa ser convenientemente controlada com o objetivo de comparar o desempenho entre os métodos de aprendizado multirrótulo.

Contents

1 Introduction

This chapter presents an overview of this thesis. It provides an introduction to the main issues studied in Multi-label Learning (MLL), an important machine learning scenario where objects are associated with multiple class labels simultaneously. MLL has attracted attention from many fields, such as text classification, functional genomics, image annotation and music categorization. The study of the MLL, although present for more than 20 years (MCCALLUM, 1999), only recently has gained focus on the dependence among labels.

In Section 1.1, the motivation of the thesis is presented as well as the context in which it is inserted. Section 1.2 is dedicated to show the bibliography produced during the doctorate. The chapter ends with Section 1.3, summarizing the structure of the rest of this thesis.

1.1 Context and Motivation

According to Michalski, Bratko e Bratko (1998) machine learning is a field of artificial intelligence whose objective is the development of algorithms related to learning as also the development of systems capable of automatically acquiring knowledge. One of its sub-fields is supervised learning where algorithms seek to learn a concept based on a dataset composed of known objects. The concept being learned is the relation between the features of an object and its classes or categories. Take as an example the problem of discriminating the type of flower based on the length and width of its petals and sepals. Here, the flower is the object of the problem, the length and width of its petals and sepals are its main features and the type of flower is the class. As it can be seen, the objects being studied (also called instances) in this field are usually represented by a set of features (relevant aspects of an object) and by a set of classes. To learn the desired concepts, the algorithm uses training examples whose classes are already known. The class, also called label, plays a decisive role in supervised learning. In many situations, the instance can only be associated to a single label. In many other situations, the instance can be associated within multiple labels simultaneously. This thesis works with the latter case where the problem is formally defined as Multi-Label Learning (MLL) (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010). MLL is well-known for the extensive number of label combinations a single object can have, which is exponential with respect to the number of possible labels. MLL problems appear in many fields, relevant works are found in bioinformatics, medical diagnosis, image recognition and specially in text categorization (TSOUMAKAS; KATAKIS, 2007; CARVALHO; FREITAS, 2009).

One of the main issues in MLL is the analysis of the impacts of dependencies among labels (DEMBCZYŃSKI et al., 2012; DEMBCZYŃSKI et al., 2013). Although theoretical observations about the impacts are desired, they are present in just a few studies due to the high difficulty that comes from the large number of variables to be considered (DEMBCZYŃSKI; CHENG; HÜLLERMEIER, 2010; DEMBCZYŃSKI et al., 2012; WAEGEMAN et al., 2014; MELLO et al., 2019). In order to proceed with a theoretical analysis in MLL, a mathematical formulation of the problem it is highly desired, such that it enables researches to easily apply mathematical tools of proofing. For this purpose, Dembczyński, Cheng e Hüllermeier (2010) elaborated a probabilistic framework in which concepts in MLL are mathematically formulated. Several concepts and variables in MLL are abstracted, supporting the elaboration of analyses and formal tests, without loss of generalization. Therefore, this framework is adopted in this thesis as a base formulation of MLL. Even so, finding mathematical proofs are still a challenging task and useful in this field. This motivates Chapter 3 and Chapter 4.

When focusing on the impacts of the dependence among labels, if no mathematical proof is presented, studies usually present conclusions based on experiments on artificial datasets instead (TOMÁS et al., 2014; NOH; SONG; PARK, 2004). Generally, the analysis consists of testing several multi-label algorithms in two different types of artificially generated databases, one with a high dependency between labels and the other with a low dependency. Based on the results of the performance of the algorithms, several conclusions are drawn. This work highlights one problem with these analyses: the absence of a metric for quantifying the intensity of the dependence among labels. Without this measure, it is not easy to compare methods and results from distinct authors since it is hard to know whether a dataset created by an author has the same level of dependence among labels as in a dataset of another author. This motivates this work for defining a measure that quantifies the level of dependence among labels of a probability distribution, alongside with a framework for using it to understand the impacts of label dependence on multi-label algorithms. The details of this work are found in Chapter 5.

1.2 Bibliographic production

This section lists papers produced for journals and conferences during the doctoral period. From a total of four works produced, one was published in an international journal, one was published in an international conference, and two are in the process of revision in international journals. References to the papers are listed below.

- MELLO, L. H. S. et al. NP-hardness of minimum expected coverage. Pattern Recognition Letters, v. 117, p. 45 – 51, 2019. (MELLO et al., 2019)

- MELLO, L. H. S. et al. Metric learning for electrical submersible pump fault diagnosis. In:International Joint Conference on Neural Networks 2020. 2020. (MELLO et al., 2020)

- MELLO, L. H. S.; VAREJÃO, F. M.; RODRIGUES, A. L. An experimental framework for evaluating loss minimization in multi-label classification via stochastic process. Knowledge-Based Systems journal. **submetido à publicação**.

- MELLO, L. H. S.; VAREJÃO, F. M.; RODRIGUES, A. L. A worst case analysis of calibrated label ranking multi-label classification method. Journal of Machine Learning. **submetido à publicação**.

1.3 Structure

The content of this thesis starts with a literature review of multi-label learning (MLL) in Chapter 2, where a formal definition and a statistical perspective of MLL are given. The contributions of this thesis are divided in Chapters 3, 4 and 5. Each chapter presents a distinct way of tackling the problem of analysing the impacts of label dependence in MLL, and, in the end, draws its own conclusions about its results.

Chapter 3 shows that the presence of label dependence makes an optimization of a specific MLL problem NP-HARD, even in a restricted scenario. Chapter 4 presents mathematical proofs with respect to the performance of two very related multi-label methods, calibrated label ranking and ranking by pairwise comparison, in the worst case scenarios with high and low label dependence. In Chapter 5 an experimental framework based on a stochastic process with the main purpose of measuring quantitatively the effects of label dependence on the performance of various multi-label algorithms is shown. After conclusions are drawn at the end of each chapter, the final chapter (Chapter 6) presents overview conclusions with respect to the thesis as a whole.

2 Multi-label Learning

This chapter presents the concepts and literature review about multi-label learning that are useful for understanding the thesis. When referring to Multi-Label Learning (MLL), authors often refer to a field that contains Multi-Label Classification (MLC) and label ranking (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010). In MLC the objective is to predict the labels that an object belongs to while in Label Ranking, the objective is to produce a ranking of all labels. Note that label ranking produces a preference among all labels while MLC produces a bipartition where preferences are made only from one group to another. They are also different in the number of possible solutions: In label ranking, there are $n!$ possible rankings, while in MLC, there are 2^n.

This chapter includes material from MLC, MLR, loss minimization, multi-label metrics and multi-label algorithms. It starts with a formal definition and a statistical perspective of MLL and then proceeds to describe all common multi-label methods, in Section 2.1, and all common multi-label metrics, in Section 2.2. And finally, Section 2.3 consists of information on the optimization of multi-label metrics.

Let $\mathcal{X}$ denote a feature space and $\mathcal{L} = \{\ell_1, \ell_2, \ell_3, ...\ell_n\}$ be a set of labels with $n = |\mathcal{L}|$. An instance is defined as a pair of two vectors $(\mathbf{x}, \mathbf{y})$ where $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y}$ is a labelling (combination of labels) represented by binary vector $\mathbf{y} = (y_1, y_2, ..., y_n)$ such that $y_i = 1$ only if the respective instance is associated to label ℓ_i. Let $\mathcal{Y} = \{0, 1\}^n$ denote the set of all possible labellings of n labels.

It is assumed that labellings are distributed according to a conditional probability distribution $\mathbf{P}(\mathbf{Y}|\mathbf{X})$ where $\mathbf{X}$ is a random vector defined in $\mathcal{X}$ and $\mathbf{Y}$ is a random vector defined on $\mathcal{Y}$. This means that for a specific feature vector $\mathbf{x} \in \mathcal{X}$, each labelling $\mathbf{y} \in \mathcal{Y}$ occurs with a probability of $\mathbf{P}(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x})$ [1]. For simplification, the given feature vector will be omitted. Therefore, in the rest of the work, assume $\mathbf{x}$ is always given. Also let $\mathbf{P}^{(i)}$, for an arbitrary distribution $\mathbf{P}$ of $\mathbf{Y}$, be defined as the marginal distribution of label i, that is,

$$\mathbf{P}^{(i)} = \sum_{\mathbf{y} \in \mathcal{Y}: y_i = 1} \mathbf{P}(\mathbf{Y} = \mathbf{y}).$$

The label cardinality of a multi-label problem (or dataset) is the average number of relevant labels per object/instance, i.e, $\sum_{i=1}^{n} \mathbf{P}^{(i)}$, while the label density is the label cardinality divided by n: $\frac{1}{n} \sum_{i=1}^{n} \mathbf{P}^{(i)}$.

The risk of a multi-label method $\mathbf{h}$ and feature vector $\mathbf{x} \in \mathcal{X}$ is defined as the

[1] $\mathbf{P}(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x})$ stands for the conditional probability that an instance has labelling $\mathbf{y}$, given its feature vector $\mathbf{x}$.

conditional expected loss given by

$$\begin{aligned} R_L(\mathbf{h}, \mathbf{X}) &= \mathbb{E}_{\mathbf{Y}|\mathbf{X}} L(\mathbf{Y}, \mathbf{h}(\mathbf{X})) \\ &= \sum_{\mathbf{y} \in \mathcal{Y}} L(\mathbf{y}, \mathbf{h}(\mathbf{X})) \cdot \mathbf{P}(\mathbf{Y} = \mathbf{y}|\mathbf{X}), \end{aligned} \tag{2.1}$$

where $\mathbf{P}(\mathbf{Y} = \mathbf{y}|\mathbf{X})$ represents the conditional probability of $\mathbf{Y}$ given a feature vector $\mathbf{X}$ and $L(\cdot)$ is a loss function on multi-label predictions.

The regret of a multi-label method $\mathbf{h}$ with respect to a loss function L is defined as

$$r_L(\mathbf{h}, \mathbf{x}) = R_L(\mathbf{h}, \mathbf{x}) - R_L(\mathbf{h}^*, \mathbf{x}), \tag{2.2}$$

where $\mathbf{h}^*$ is a Bayes-optimal method that yields the minimum loss for L. The feature vector will be omitted in the rest of this thesis. In regret, it is common the idea of $\mathbf{h}$ making mistakes with respect to $\mathbf{h}^*$. The idea of comparing $\mathbf{h}$ to $\mathbf{h}^*$ and associating the differences among them as mistakes, is a useful concept for the regret analysis. If it is a MLC problem, then a mistake would be $h_i \neq h_i^*$. Analogously, if it is a label ranking problem, then let be defined the misorder of a rank $\mathbf{z}$ on a pair of labels (i, j) with respect to an optimal rank $\mathbf{z}^*$ when $z_i > z_j$ and $z_i^* < z_j^*$.

In most practical cases, the distribution $\mathbf{P}$ is unknown. However, sometimes, especially in theoretical analysis, the distribution of labels $\mathbf{P}$ is assumed to be known and given. In these cases, the risk at (2.1) can be redefined to

$$R_L(\mathbf{h}, \mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} L(\mathbf{y}, \mathbf{h}(\mathbf{P})) \cdot \mathbf{P}(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}), \tag{2.3}$$

where $\mathbf{h}$ is a function that predicts a labelling based on the distribution of labels. In this scenario, multi-label method $\mathbf{h}$ from (2.1) and multi-label method $\mathbf{h}$ from (2.3) are essentially different, but can share similar ideas or computations. This difference and similarities are discussed in the next section.

2.1 Multi-label methods

A multi-label method $\mathbf{h}$ can be a multi-label classifier or a multi-label ranker. A multi-label classifier predicts a labelling, in which $\mathbf{h} : \mathcal{X} \to \mathcal{Y}$, and for a given instance $\mathbf{x} \in \mathcal{X}$ it returns a vector $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), ..., h_n(\mathbf{x}))$, where h_i represents the presence/absence of label i. A multi-label ranker $\mathbf{h}$ predicts a ranking, in which $\mathbf{h} : \mathcal{X} \to \mathfrak{S}_n$, where $\mathfrak{S}_\mathfrak{n}$, representing the set of all possible rankings, is the set of all permutations of $\{1, ..., n\}$. In this case, $h_i(\mathbf{x})$ denotes the rank of label i for a given instance $\mathbf{x} \in \mathcal{X}$.

A binary classifier is usually defined as a function that predicts positive/negative classes given an observation $f : \mathcal{X} \mapsto \{0, 1\}$. However, some binary classifiers can also predict an estimate of the probability distribution of the target label. They are called probabilistic classifiers and usually they estimate the conditional probability $\mathbf{P}(Y = 1|\mathbf{X} = \mathbf{x})$ for some fixed label Y. Consequently, they can be written as a function from the feature space to a probability value: $g : \mathcal{X} \mapsto [0, 1]$. For these classifiers, the final prediction can be rewritten as a function of the estimated conditional probability: $h : [0, 1] \mapsto \{0, 1\}$, so that, $f(\mathbf{x}) = h(g(\mathbf{x}))$. For instance, g can be a trained logistic regression classifier of the form

$$g(\mathbf{x}) = \frac{1}{1 + e^{-\theta \cdot \mathbf{x}}},$$

where $\theta \in \mathbb{R}^n$ are trained parameters, and h can be a threshold function of the form

$$h(\mathbf{P}) = \begin{cases} 1, & \text{if } \mathbf{P}(Y = 1|\mathbf{X} = \mathbf{x}) > \lambda \\ 0, & \text{otherwise,} \end{cases}$$

where $\lambda \in [0, 1]$ is a desired threshold value (usually 1/2) that can also be estimated. Note that $g(\mathbf{x})$ is an estimate of the conditional probability and, therefore, can propagate errors to the prediction of the label presence/absence, even if h is a perfect optimized function.

Analogously, one can define a probabilistic multi-class classifier as $h(\mathbf{g}(\mathbf{x}))$ where $\mathbf{g} : \mathcal{X} \mapsto [0, 1]^n$ represents a mapping to n probability values $\mathbf{P}(Y_i = 1|\mathbf{x})$, one for each label, and $h : [0, 1]^n \mapsto \mathcal{L}$ represents the prediction of a label in $\mathcal{L}$. Also, one can define a probabilistic multi-label classifier as $\mathbf{h}(\mathbf{g}(\mathbf{x}))$ where $\mathbf{g} : \mathcal{X} \mapsto [0, 1]^{2^n}$ represents a mapping to 2^n probability values $\mathbf{P}(\mathbf{Y} = \mathbf{y}|\mathbf{x})$, one for each combination of labels $\mathbf{y} \in \mathcal{Y}$, and $\mathbf{h} : [0, 1]^{2^n} \mapsto \mathcal{Y}$ represents the actual prediction of a labelling in $\mathcal{Y}$. This is illustrated in Figure 1, where an induced model first produces an estimate of the label probability distribution, and then a predicted labelling is made. For instance, one can use the Label Powerset strategy (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010; ZHANG; ZHOU, 2013): transform a multi-label problem of n labels into a multi-class problem of 2^n classes, where each class represents a combination of labels in $\mathcal{Y}$, and then using any probabilistic multi-class classifier $\mathbf{g}$ to estimate the whole joint probability distribution. Function $\mathbf{h}$ can then be the mode of the estimated probability distribution:

$$\mathbf{h}(\mathbf{P}) = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} \mathbf{P}(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}).$$

In practice, many probabilistic multi-label classifiers do not use the whole probability distribution $\mathbf{P}(\mathbf{Y}|\mathbf{X} = \mathbf{x})$ of 2^n values, so they only estimate part of it. Details and algorithms about estimating the whole label distribution are presented by Dembczyński, Cheng e Hüllermeier (2010), Geng (2016), Jia et al. (2018), Wang e Geng (2019), Sun e Kudo (2018), where authors present and evaluate algorithms for learning label distributions.

In the present thesis, only the second part (function $\mathbf{h}$) is taken into consideration when analysing a specific multi-label method. The rest of this section defines the second part (function $\mathbf{h}$) of five multi-label classifiers: Binary Relevance (BR)(TSOUMAKAS;

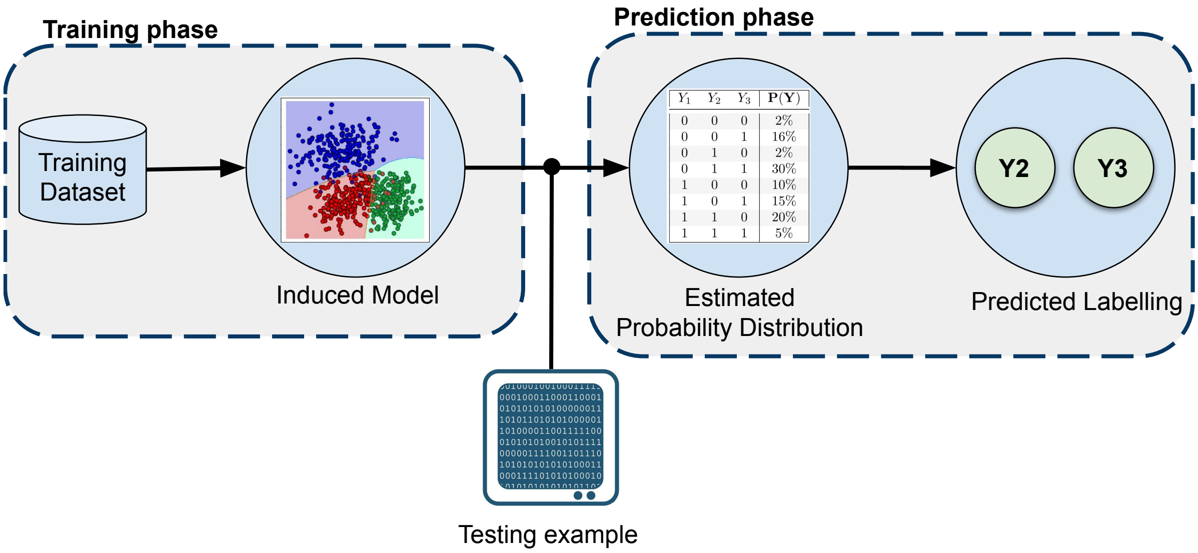


Figure 1 – Illustration of a multi-label probabilistic classifier. The training phase is responsible for building an induced model based on a given training dataset, while the prediction phase is responsible for predicting a labelling based on a label probability distribution of a giving testing example.

KATAKIS, 2007), Classifier Chains (CC)(READ et al., 2009), Dependent Binary Relevance (DBR)(MONTAÑES et al., 2014), Probabilistic Classifier Chains (PCC)(DEMBCZYŃSKI; CHENG; HÜLLERMEIER, 2010) and Calibrated Label Ranking (CLR) (FÜRNKRANZ et al., 2008).

2.1.1 Binary Relevance

A widely used transformation method is the Binary Relevance (BR) approach (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010), in which a binary classifier is trained for each label independently. It transforms the original problem into n binary classification problems where the positive label in the i-th problem is the label i while all other labels are viewed as negative. At the end, BR predicts a label as positive if its corresponding binary classifier gives a positive output. Therefore, from a probabilistic perspective, the prediction done by BR for label y_i, call it y_i^{BR}, is given by:

$$y_i^{\mathrm{BR}} = \underset{\ell \in \{0,1\}}{\operatorname{argmax}} \; \mathbf{P}(Y_i = \ell). \tag{2.4}$$

2.1.2 Dependent Binary Relevance

Other well-known transformation method is The Dependent Binary Relevance (DBR) (MONTAÑES et al., 2014) extends BR to consider label dependencies. It first predicts a labelling in the same way as BR, then it uses n new binary classifiers, one for each label, to predict a new labelling based on the labelling given by BR. The i-th binary classifier assumes that all labels given by BR are all correct, except for the label i which may be changed given the new information (all other labels). Hence, the prediction done by DBR is given by

$$y_i^{\text{DBR}} = \operatorname*{argmax}_{\ell \in \{0,1\}} \mathbf{P}(Y_i = \ell | \bigwedge_{1 \leq j \leq m : j \neq i} Y_j = y_j^{\text{BR}}).$$

2.1.3 Classifier Chains

Inspired by the simplicity and computational efficiency of the BR method, The Classifier Chains (CC) method also transforms the original problem into n binary classification problems, each one with its corresponding binary classifier. Unlike BR, the binary classifiers composing CC do not predict independently. These binary classifiers are organized in a chain (randomly defined) such that the i-th binary classifier assumes the output prediction of all its previous classifiers are correct and uses them to make its own prediction. Hence, the prediction done by CC is given by

$$\begin{aligned} y_1^{\text{CC}} &= y_1^{\text{BR}} \\ y_i^{\text{CC}} &= \operatorname*{argmax}_{\ell \in \{0,1\}} \mathbf{P}(Y_i = \ell | \bigwedge_{1 \leq j < i} y_j = y_j^{\text{CC}}), \ i \geq 2. \end{aligned}$$

The purpose of chaining classifiers and their output is to consider label dependencies in order to achieve better performance, while maintaining the simplicity and computational cost of BR.

2.1.4 Probabilistic Classifier Chains

The method Probabilistic Classifier Chains (PCC) (DEMBCZYŃSKI; CHENG; HÜLLERMEIER, 2010) is an expansion of the CC. Instead of using only the joint probability of a single labelling $\mathbf{y}$ as CC, PCC uses the probability of all 2^n possible labellings and then predicts the most probable. Therefore, the prediction done by PCC is given by the mode of the label distribution:

$$\mathbf{y}^{\text{PCC}} = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} \mathbf{P}(\mathbf{Y} = \mathbf{y}).$$

The PCC method estimates the label distribution by using the product rule of probability and an augmented input feature space:

$$\begin{aligned} \mathbf{P}_{\mathbf{x}}(\mathbf{Y} = \mathbf{y}) &= \mathbf{P}_{\mathbf{x}}(Y_1 = y_1) \cdot \prod_{i=2}^{n} \mathbf{P}_{\mathbf{x}}(Y_i = y_i | Y_1, ..., Y_{i-1}), \\ &\approx g_1(\mathbf{x}) \cdot \prod_{i=2}^{n} g_i(\mathbf{x}, y_1, ..., y_{i-1}), \end{aligned}$$

where $g_i : \mathcal{X} \times \{0,1\}^{i-1}$ is the augmented input feature space and it takes $y_1, ..., y_{i-1}$ as additional features.

2.1.5 Label Powerset

Like PCC, the Label Powerset method (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010) also predicts the mode of the label distribution, since it predicts the combination of labels with the highest probability. Both these algorithms computes their prediction in exponential time $\mathcal{O}(2^n)$. The difference between them lies on how they estimate the label distribution. While PCC uses an augmented input feature space, Label Powerset transforms each combination of labels into a distinct class and then uses a probabilistic multi-class classifier.

2.1.6 RAKEL

RAKEL (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010) is a multi-label method inspired in the Label Powerset method. Defining RAKEL is easier when using the set definition of a labelling ($\ell \subseteq \mathcal{L}$) than using the binary vector definition ($\mathbf{y} \in \mathcal{Y}$). Given an integer k divisor of n as parameter[2], RAKEL defines a random partition of n/k subsets $\mathcal{L}_1 \cup ... \cup \mathcal{L}_{n/k} = \mathcal{L}$ such that $|\mathcal{L}_i| = k$ for all i. Then the prediction of RAKEL, given the probability distribution of labels, is

$$\begin{aligned} L^{\text{rakel}} = \underset{\ell \subseteq \mathcal{L}_1}{\operatorname{argmax}} \mathbf{P}(\mathbf{L} = \ell | \mathbf{X} = \mathbf{x}) \quad & \cup \quad \underset{\ell \subseteq \mathcal{L}_2}{\operatorname{argmax}} \mathbf{P}(\mathbf{L} = \ell | \mathbf{X} = \mathbf{x}) \quad \cup \quad ... \\ & \cup \quad \underset{\ell \subseteq \mathcal{L}_{n/k}}{\operatorname{argmax}} \mathbf{P}(\mathbf{L} = \ell | \mathbf{X} = \mathbf{x}) \end{aligned}$$

Note that for $k = 1$ RAKEL is identical to BR, and for $k = n$ it is identical to Label Powerset.

2.1.7 F-measure optimizer

An efficient exact algorithm for optimizing F-measure is given by Dembczyński et al. (2013), where n^2 parameters of the probability label distribution are used. Let p_{ik}, for $1 \leq i \leq n$ and $1 \leq k \leq n$, be defined as

$$p_{ik} = \sum_{s=1}^{n} \frac{\mathbf{P}(Y_i = 1 | \mathbf{S} = s)}{s + k},$$

where $\mathbf{S} = \sum_{i=1}^{n} Y_i$, i.e, the number of positive labels. $\mathbf{P}(Y_i = 1 | \mathbf{S} = s)$ can be read as the probability of having the i-th label, giving that there are s positive labels. Sort labels such that $p_{11} \geq p_{21} \geq p_{31} \geq \cdots \geq p_{n1}$. The number of positive labels in the prediction of the optimizer is given by k^*:

$$k^* = \underset{k \in \{0..n\}}{\operatorname{argmax}} \; p_k, \tag{2.5}$$

[2] For simplicity, parameter k was assumed to be a divisor of n. The equation regarding RAKEL can be easily generalized to any integer $1 \leq k \leq n$.

where

$$p_k = \begin{cases} \sum_{i=1}^{k} p_{ik}, & \text{if } k \geq 1, \\ \mathbf{P}(\mathbf{Y} = 0), & \text{if } k = 0. \end{cases}$$

Finally, the prediction of the optimizer is given by making the first k^* labels positive (after the sort above) and the others negative.

2.1.8 Ranking by pairwise comparison

Ranking by pairwise comparison (RPC) is a multi-label method composed of $\frac{n(n-1)}{2}$ binary classifiers, with the purpose of building a ranking for a given instance. The ranking is built by first giving a score s_i for each label i. The score is computed by a pairwise preference scheme where there exists a binary classifier for each distinct pair of labels (say i and j) whose task is to distinguish the occurrence of label i and label j when assuming that only one of both occurs. Therefore, each classifier outputs its preference towards one of the two labels. A pseudo code for training RPC is presented in Algorithm 1 and the computation of the score of a single label is presented in Algorithm 2.

```
Data : Training data set of $m$ samples $\mathbf{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_m, \mathbf{y}_m)\}$
Result : Trained binary classifiers $c_{ij}$ for $1 \leq i \leq n$, $1 \leq j \leq n$ and $i \neq j$.
for each pair of labels $i, j$ do
    $\mathbf{D}' := \{(\mathbf{x}, \mathbf{y}) \in \mathbf{D} : y_i = 1 \text{ and } y_j = 0\}$
    $\mathbf{D}' := \{(\mathbf{x}, 1) : (\mathbf{x}, \mathbf{y}) \in \mathbf{D}'\}$    // replace all labellings with a 1, representing the positive class.

    $\mathbf{D}'' := \{(\mathbf{x}, \mathbf{y}) \in \mathbf{D} : y_i = 0 \text{ and } y_j = 1\}$
    $\mathbf{D}'' := \{(\mathbf{x}, 0) : (\mathbf{x}, \mathbf{y}) \in \mathbf{D}''\}$    // replace all labellings with a 0, representing the negative class
    $c_{ij} :=$ train_binary_classifier($\mathbf{D}' \cup \mathbf{D}''$)    // Binary classification problem.
end
```

Algorithm 1 : Algorithm for training RPC.

```
Input : Trained binary classifiers $c_{ij}$ for all $j \neq i$.
Result : Score $s \in \mathbb{N}$
$s := 0$
for each label $j$ different of $i$ do
    $\ell =$ predict_label($c_{ij}$, $\mathbf{x}$)    // Function predict_label returns 1 if $i$ is predicted positive, otherwise 0.
    $s := s + \ell$    // +1 if $i$ is predicted positive by $c_{ij}$.
end
```

Algorithm 2 : Scoring a single label i in RPC.

Given this definition, let RPC be defined as a ranking method that prefers label i to label j if $s_i > s_j$, where s_i is computed by

$$s_i = \sum_{k \neq i} [\![\mathbf{P}(Y_i = 1, Y_k = 0 | Y_i = 1 \oplus Y_k = 1) > 0.5]\!],$$

where $(Y_i = 1 \oplus Y_k = 1)$ means $Y_i = 1$ or $Y_k = 1$ exclusively and $[\![e]\!]$ is the Iverson brackets, which evaluates to 1 if expression e is true, 0 otherwise. The probability is conditioned on $Y_i = 1 \oplus Y_k = 1$, because in Algorithm 1, the binary classifier c_{ij} is trained on $\mathbf{D}' \cup \mathbf{D}''$ (Line 7), which is equivalent to $\{(\mathbf{x}, \mathbf{y}) \in \mathbf{D} : y_i = 1 \oplus y_j = 1\}$, but replacing all labellings $\mathbf{y}$ with a 1 when $y_i = 1$, and with a 0 when $y_j = 1$. Therefore, the value $[\![\mathbf{P}(Y_i = 1, Y_k = 0 | Y_i = 1 \oplus Y_k = 1) > 0.5]\!]$ corresponds to the vote given by the binary classifier responsible for distinguishing the presence of label pair (i, k). It is worth mentioning that $\mathbf{P}(Y_i = 1, Y_k = 0 | Y_i = 1 \oplus Y_k = 1)$ can be rewritten as:

$$\mathbf{P}(Y_i = 1, Y_k = 0 | Y_i = 1 \oplus Y_k = 1) = \frac{\mathbf{P}(Y_i = 1, Y_k = 0)}{\mathbf{P}(Y_i = 1, Y_k = 0) + \mathbf{P}(Y_i = 0, Y_k = 1)},$$

which is sometimes a more convenient form for calculating this conditional probability.

There may exist cases in which $Y_i = 1 \oplus Y_k = 1$ never occurs. In practice, this would mean that the binary classifier responsible for distinguishing label i from k would be trained on an empty dataset. In this case, usually a value from $\{0, \frac{1}{2}, 1\}$ ($\frac{1}{2}$ is the most frequent choice) is arbitrarily adopted for $[\![\mathbf{P}(Y_i = 1, Y_k = 0 | Y_i = 1 \oplus Y_k = 1) > 0.5]\!]$. Whatever the choice, as long as,

$$\mathbf{P}(Y_i = 1, Y_k = 0 | Y_i = 1 \oplus Y_k = 1) + \mathbf{P}(Y_i = 0, Y_k = 1 | Y_i = 1 \oplus Y_k = 1) = 1,$$

is satisfied, which is already true for $\mathbf{P}(Y_i = 1 \oplus Y_k = 1) \neq 0$, the proofs in this thesis are valid.

2.1.9 Calibrated Label ranking

Calibrated label ranking (CLR) is an adaptation of RPC for MLC. It adds an artificial label for constructing a bi-partition (a.k.a classification). The score of the artificial label is given by n binary classifiers that are identical to the n binary classifiers of BR method, as pointed out by Fürnkranz et al. (2008). The artificial label represents the "negative label" inside the one-against-all strategy of BR. A label is said to be positive or relevant if the score s_i, as defined above, is greater than the score of the artificial label. Note that now the score s_i should also include the artificial label. Therefore, CLR is a classifier that predicts label i as positive only if

$$\sum_{k \neq i} [\![\mathbf{P}(Y_i = 1, Y_k = 0 | Y_i = 1 \oplus Y_k = 1) > 0.5]\!] + [\![\mathbf{P}(Y_i = 1) > 0.5]\!] > \sum_{k=1}^{n} [\![\mathbf{P}(Y_k = 0) > 0.5]\!].$$

The summation on the right-hand side of the inequality counts the number of votes in favor of the calibrated/artificial label and $[\![\mathbf{P}(Y_i = 1) > 0.5]\!]$ corresponds to the vote given by a one-against-all classifier (see the BR method). Observe that, although CLR is trained on $Y_i = 1 \oplus Y_k = 1$, the algorithm can output multiple positive labels. It will usually output multiple positive labels if $\sum_{k=1}^{n}[\![\mathbf{P}(Y_k = 0) > 0.5]\!]$ is low, i.e, if the label cardinality is high.

Although the name CLR is often used in the literature to describe its ranking and/or classification components, in this document the name RPC will be used to emphasize the ranking component while CLR to emphasize its multi-label classification component. For the sake of simplicity, define function $f(\mathbf{P}, i, j)$ as

$$f(\mathbf{P}, i, j) = \begin{cases} \mathbf{P}(Y_i = 1, Y_j = 0 | Y_i = 1 \oplus Y_j = 1), & \text{if } i \neq j \\ 0, & \text{if } i = j, \end{cases}$$

so that, the CLR prediction of label i can be redefined as:

$$\sum_{j=1}^{n}[\![f(\mathbf{P}, i, j) > 0.5]\!] \ + \ [\![\mathbf{P}(Y_i = 1) > 0.5]\!] > \sum_{j=1}^{n}[\![\mathbf{P}(Y_j = 0) > 0.5]\!],$$

and the RPC preference of i over j can be redefined as:

$$\sum_{k=1}^{n}[\![f(\mathbf{P}, i, k) > 0.5]\!] > \sum_{k=1}^{n}[\![f(\mathbf{P}, j, k) > 0.5]\!].$$

The versions where both CLR and RPC use the probability values as weights for voting are respectively expressed as

$$\sum_{j=1}^{n} f(\mathbf{P}, i, j) \ + \ \mathbf{P}(Y_i = 1) > \sum_{j=1}^{n} \mathbf{P}(Y_j = 0), \tag{2.6}$$

and

$$\sum_{k=1}^{n} f(\mathbf{P}, i, k) > \sum_{k=1}^{n} f(\mathbf{P}, j, k). \tag{2.7}$$

Note that the scores given by RPC and CLR to label i are respectively defined as

$$s_i = \sum_{k=1}^{n} f(\mathbf{P}, i, k), \tag{2.8}$$

and

$$s_i = \sum_{j=1}^{n} f(\mathbf{P}, i, j) \ + \ \mathbf{P}(Y_i = 1). \tag{2.9}$$

2.2 Multi-label metrics

Multi-label metrics are used to quantify the quality of predictions or the cost for inaccuracy of predictions. When a metric quantifies the error, it is called a loss function, otherwise it is called a utility function. In multi-label learning, a loss function is a function $L(\cdot)$ of the target labelling $\mathbf{y}$, which is the correct labelling, and the predicted output of a multi-label method $\mathbf{h}$. The loss function $L(\cdot)$ associates a cost to the prediction. As defined in Section 2.1, the predicted output of a multi-label method can be a labelling, denoted by $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_n)$, or it can be a ranking, denoted by $\mathbf{z} = (z_1, z_2, ..., z_n)$, where z_i is the ranking of label i. Therefore, a classification based loss function is defined as a mapping $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ for classification and ranked based loss function $L : \mathcal{Y} \times \mathfrak{S} \to \mathbb{R}^+$, where $\mathfrak{S}$ is the set of all permutations of $\{1, ..., n\}$. The metric Hamming loss is a function of a classification which is defined as the fraction of labels whose presence is incorrectly predicted:

$$L_H(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} [\![y_i \neq \hat{y}_i]\!]$$

where $[\![e]\!]$ is the Iverson brackets, evaluating to 1 if expression e is true, 0 otherwise. Other common loss function is the subset 0/1 loss, which detects a strict coincidence of the actual and estimated labels as

$$L_s(\mathbf{y}, \hat{\mathbf{y}}) = [\![\mathbf{y} \neq \hat{\mathbf{y}}]\!].$$

More elaborate loss functions are the loss version of the F-measure and the Jaccard distance given respectively by

$$L_f(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{2 \sum_{i=1}^{n} y_i \hat{y}_i}{\sum_{i=1}^{n} (y_i + \hat{y}_i)}$$

and

$$L_j(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\sum_{i=1}^{n} y_i \hat{y}_i}{\sum_{i=1}^{n} (y_i + \hat{y}_i) - \sum_{i=1}^{n} y_i \hat{y}_i}.$$

Note that the formula for both F-measure loss and Jaccard distance are quite similar. Indeed, both metrics share a close relation as one can be obtained from the other:

$$L_j(\mathbf{y}, \hat{\mathbf{y}}) = \frac{2\, L_f(\mathbf{y}, \hat{\mathbf{y}})}{1 + L_f(\mathbf{y}, \hat{\mathbf{y}})}.$$

Since $L_f(\mathbf{y}, \hat{\mathbf{y}}) \leq 1$, it follows that

$$L_j(\mathbf{y}, \hat{\mathbf{y}}) \geq L_f(\mathbf{y}, \hat{\mathbf{y}}), \quad \forall \mathbf{y}, \hat{\mathbf{y}} \in \mathcal{Y}. \tag{2.10}$$

A simple metric that takes into account a rank is the rank loss, which is defined as

$$L_r(\mathbf{y}, \hat{\mathbf{z}}) = \sum_{(i,j): y_i > y_j} [\![\hat{z}_i < \hat{z}_j]\!].$$

The normalized rank loss is defined as

$$L_{\hat{r}}(\mathbf{y}, \hat{\mathbf{z}}) = \frac{L_r(\mathbf{y}, \hat{\mathbf{z}})}{s_{\mathbf{y}}(n - s_{\mathbf{y}})},$$

where $s_{\mathbf{y}} = \sum_{i=1}^{n} y_i$.

These metrics are the most common in the multi-label scenario, and they will be used to analyze the performance of CLR. Two metrics from the preference learning field are the squared rank distance

$$L_{\text{srd}}(\mathbf{z}, \hat{\mathbf{z}}) = \sum_{i}^{n} (z_i - \hat{z}_i)^2$$

and the Spearman rank correlation[3] (Hüllermeier; Furnkranz, 2004). The Spearman rank correlation is defined as the Pearson correlation between the ranked values of two variables. In the context of preference learning, the Spearman rank correlation can be obtained by the following formula

$$1 - \frac{6L_{\text{srd}}(\mathbf{z}, \hat{\mathbf{z}})}{n(n^2 - 1)}.$$

The Spearman rank correlation can be interpreted as a linear normalization of the squared rank distance to the interval $[-1, 1]$.

The loss function named coverage (SCHAPIRE; SINGER, 2000) is another that takes a rank and a labelling into account, and we define as the rank of the last relevant label in the ranked list, i.e the rank of the label with the highest rank that is relevant:

$$L_c(\mathbf{y}, \mathbf{z}) = \max_{i:y_i=1} (z_i). \tag{2.11}$$

The special case in which there is no relevant label (i.e $\mathbf{y} = 0_n$), coverage evaluates to 0. Usually, coverage is defined as our above definition, but 1 subtracted, i.e $L_c(\mathbf{y}, \mathbf{z}) - 1$. The reason behind the minus 1 is that they assume labellings with all zeroes (i.e, no relevant label) do not occur and then subtract 1 so that the minimum possible coverage is zero. The search length (CHEN; KARGER, 2006) is similar to coverage, and it is defined as the rank of the first (instead of the last) relevant label in the ranked list. Despite the fact that it is commonly used for document ranking (COOPER, 1968; CHEN; KARGER, 2006), it can also be used for multi-label learning. It is formally defined as

$$L_\ell(\mathbf{y}, \mathbf{z}) = \min_{i:y_i=1} (z_i). \tag{2.12}$$

The special case in which there is no relevant label (i.e $\mathbf{y} = 0_n$), search length evaluates to 0.

2.3 Optimal Risk and Regret

This section presents a brief review and definitions on optimal solutions for the risk minimization and the regret of some metrics defined in Section 2.2. These definitions will be quite useful for Chapter 4 and Chapter 5.

[3] Spearman rank correlation in the preference learning and multi-label ranking field is a utility function. The higher, the better.

The task of risk minimization is finding a model $\mathbf{f}^*$ that minimizes function R for Equation (2.1):

$$\mathbf{f}^* = \operatorname*{argmin}_{\mathbf{f}} R_L(\mathbf{f}, \mathbf{x}).$$

The $\operatorname{argmin}_{\mathbf{f}}$ is not "friendly", since it consider the universe of all functions of $\mathbf{x}$ mapping to a labelling. If the probability distribution $\mathbf{P}$ is given, as assumed in Equation 2.3, then the optimal solution $\mathbf{y}^*$ is much simpler:

$$\mathbf{y}^* = \operatorname*{argmin}_{\mathbf{y} \in \mathcal{Y}} R_L(\mathbf{y}, \mathbf{x}).$$

Hence, the optimal solution can clearly can be achieved by an exhaustive search, which is testing all 2^n possible labellings for classification, or testing all $n!$ rankings for label ranking, but the minimum should be obtained efficiently (in polynomial time, if possible). In general, this is NP-HARD as it contains a particular instance of risk minimization of the Jaccard distance, proved to be NP-COMPLETE (CHIERICHETTI et al., 2010). Therefore, efficient algorithms are only designed for specific metrics where specific properties can be exploited, which is the case of Hamming loss, F-measure and rank loss (DEMBCZYŃSKI et al., 2012).

The authors Cheng, Hüllermeier e Dembczyński (2010) the authors proved that the optimal labelling $\mathbf{y}^*$ for the risk of a Hamming loss can be obtained by just looking at the marginal distribution of labels, and it is given by

$$y_i^* = \begin{cases} 1, & \text{if } \mathbf{P}^{(i)} > \frac{1}{2}, \\ 0, & \text{if } \mathbf{P}^{(i)} \leq \frac{1}{2}. \end{cases} \tag{2.13}$$

The authors Dembczyński et al. (2012) have shown that one optimal labelling $\mathbf{y}^*$ for the risk of subset 0/1 loss is given by the mode of the distribution:

$$\mathbf{y}^* = \operatorname*{argmax}_{\mathbf{y}} \mathbf{P}(\mathbf{y}).$$

Interestingly, they showed that the optimal expected Hamming loss may give the worst case regret of 1/2 in subset 0/1 loss. Furthermore, the optimal expected subset 0/1 loss solution may give a regret as closely as possible to 1, with respect to Hamming loss.

The same authors Dembczyński et al. (2012) have also shown that to achieve optimal ranking $\mathbf{z}^*$ in rank loss, it is sufficient to order the labels with respect to their marginal distribution:

$$z_i^* < z_j^* \iff \mathbf{P}^{(i)} > \mathbf{P}^{(j)}.$$

Interestingly, the regret of rank loss can be obtained by just summing the difference $\mathbf{P}^{(i)} - \mathbf{P}^{(j)}$ of all pairs (i, j) with misorder (DEMBCZYŃSKI; KOTŁOWSKI; HÜLLERMEIER, 2012):

$$r_r(\mathbf{z}) = \sum_{(i,j): z_i^* < z_j^*} [\![z_i > z_j]\!] \left(\mathbf{P}^{(i)} - \mathbf{P}^{(j)} \right). \tag{2.14}$$

Regarding the normalized rank loss, let $s_{\max}$ be defined as

$$s_{\max} = \max_{\mathbf{y}:\mathbf{P}(\mathbf{y})>0} s_{\mathbf{y}}(n - s_{\mathbf{y}}),$$

it is easy to see that

$$R_{\hat{r}}(\mathbf{z}) \geq \frac{R_r(\mathbf{z})}{s_{\max}} \tag{2.15}$$

The authors Hüllermeier e Furnkranz (2004) proved that the ranking constructed by RPC is optimal for squared rank distance and, consequently, for Spearman rank correlation. Part of the pairwise approach adopted by RPC is essentially learning the preference of one label over another, which may be one of the reasons why this approach is optimal for Spearman rank correlation.

3 Optimal Risk of Coverage

It is convenient to known how to obtain the optimal solution for specific metrics to advance in the analysis of label dependencies in multi-label learning. To this end, an analysis was conducted to determine how to obtain the optimal solution for the expected value of coverage, the only metric among the chosen ones whose properties on the optimal solution are unknown. Since coverage focus on covering all relevant labels, it may be an important measure for applications requiring a low false negative value (SOROWER, 2010). For instance, coverage can be important in active learning (SETTLES, 2009), where unlabelled objects are ranked according to their relevance for being analyzed/labelled by a human specialist. For simplicity, lets say the human specialist analyze the unlabelled objects in a certain order given by an algorithm until he finds all relevant objects. If all relevant objects are ranked first, it reduces the human work by removing irrelevant objects from its analysis. Coverage is also usually used as a measure for comparing algorithms (MADJAROV et al., 2012; SCHAPIRE; SINGER, 2000; ZHANG; ZHANG, 2010). Therefore, finding an efficient and exact algorithm for minimizing coverage or proving that such an algorithm is too hard to find (NP-HARD) is a topic of scientific interest.

A simple exact algorithm, but not quite efficient, for minimizing the expected coverage is testing all possible $n!$ rankings. Unless n is pretty small, this is not tractable by a modern computer. The main concern here is to discover an algorithm capable of always computing the optimal ranking for coverage efficiently, ideally in polynomial time. Unfortunately, this is a challenging task and such algorithms were only found for specific loss functions in specific cases, such as F-measure, Hamming loss and rank loss (DEMBCZYŃSKI; CHENG; HÜLLERMEIER, 2010; DEMBCZYŃSKI et al., 2013). On the other hand, for some loss functions, it has been proved that such an algorithm does not exist unless P=NP, even for very specific cases. An example is a particular instance of risk minimization of the Jaccard distance, proved to be NP-COMPLETE (CHIERICHETTI et al., 2010).

In this present chapter a proof that optimizing the risk of coverage (a.k.a minimum expected coverage) is NP-HARD, is shown. Moreover, it shows that even assuming a very particular case where all instances have exactly two labels, computing the minimum expected coverage is still NP-HARD. Having exactly two labels is a special case of extreme multi-label classification (JASINSKA-KOBUS et al., 2020), so the results in this chapter are valuable for this field. The NP-HARDNESS remains even in a scenario where labels have a low level of dependence. Closely related problems to the proof of NP-COMPLETENESS shown in this chapter are the Document Ranking (CHEN; KARGER, 2006; ZHAI; COHEN; LAFFERTY, 2003) and the Balanced minimum sum-of-squares clustering (ARTEM;

ALOISE; MLADENOVIĆ, 2017). Both were proved to be NP-COMPLETE. Related works in the same field of MLL can be found by Dembczyński et al. (2012), where risk minimization in MLL is deeply studied, and specially by Dembczyński et al. (2013), where a polynomial algorithm is found for solving the risk minimization of a specific loss function denominated as the F-measure.

The rest of the chapter is organized in the following manner. In Section 3.1, coverage and search length are discussed and the equivalence of the maximum expected search length and the minimum expected coverage is shown. In Section 3.2, a specific case where instances have exactly two labels is proved to be NP-COMPLETE. In Section 3.3, it is given a mathematical definition of what is a scenario of a low level of dependence among labels and proved the NP-COMPLETENESS of the optimal expected coverage in this scenario.

3.1 Relationship between Search Length and Coverage

In this section, the close relationship between the risk optimization of search length (CHEN; KARGER, 2006) and coverage (SCHAPIRE; SINGER, 2000) is shown. For the sake of making some operations and equations a little easier to follow and understand, in this section and also in Section 3.2, we will use a ranking from 0 to $n-1$ instead of 1 to n. This is equivalent to using the version of coverage and search length, but with a one subtracted (see the discussion below Equation (2.11)). This does interfere in nothing about the NP-HARDNESS of the problem since subtracting an objective function by a constant does not make the function any harder/easier to optimize. A transformation composed of only simple arithmetic operations can be done on the rankings such that risk minimization of coverage becomes identical to the risk maximization of search length and vice-versa. The following propositions reveal this transformation.

Proposition 3.1. *For any ranking* $\mathbf{z}$ *and any distribution of* $\mathbf{Y}$ *given* $\mathbf{x}$*, the expected coverage is related to the expected search length by*

$$\mathbb{E}_{\mathbf{Y}|\mathbf{X}} L_c(\mathbf{Y}, \mathbf{z}) = n - 1 - \mathbb{E}_{\mathbf{Y}|\mathbf{X}} L_\ell(\mathbf{Y}, \bar{\mathbf{z}}).$$

where $\bar{\mathbf{z}}$ *is the opposite ranking of* $\mathbf{z}$*, i.e,* $\bar{\mathbf{z}}_i = n - 1 - z_i, 1 \leq \ i \ \leq \ n.$

Proof.

Considering the definition of coverage (2.11), it can be shown that

$$\begin{aligned} L_c(\mathbf{Y}, \mathbf{z}) &= \max_{i:y_i=1}(z_i) \\ &= \max_{i:y_i=1}(n-1-\bar{z}_i) \\ &= n - 1 - \min_{i:y_i=1}(\bar{z}_i), \end{aligned}$$

and so, from the definition of search length (2.12),

$$\begin{aligned} L_c(\mathbf{Y}, \mathbf{z}) &= n - 1 - \min_{i:y_i=1}(\bar{z}_i) \\ &= n - 1 - L_\ell(\mathbf{Y}, \bar{z}_i). \end{aligned}$$

□

As a corollary of Proposition 3.1, for any distribution of $\mathbf{Y}$ given $\mathbf{x}$, the minimum expected coverage is related to the maximum expected search length by

$$\min_{\mathbf{z}} \ \mathbb{E}_{\mathbf{Y}|\mathbf{X}} L_c(\mathbf{Y}, \mathbf{z}) = n - 1 - \max_{\mathbf{z}} \ \mathbb{E}_{\mathbf{Y}|\mathbf{X}} L_\ell(\mathbf{Y}, \mathbf{z}).$$

Proof.

Proposition 3.1 implies that the minimum expected coverage is given by

$$\begin{aligned} \min_{\mathbf{z}} \ \mathbb{E}_{\mathbf{Y}|\mathbf{X}} L_c(\mathbf{Y}, \mathbf{z}) &= \min_{\mathbf{z}} \Big(n - 1 - \mathbb{E}_{\mathbf{Y}|\mathbf{X}} L_\ell(\mathbf{Y}, \mathbf{z}) \Big) \\ &= n - 1 + \min_{\mathbf{z}} \Big(-\mathbb{E}_{\mathbf{Y}|\mathbf{X}} L_\ell(\mathbf{Y}, \mathbf{z}) \Big) \\ &= n - 1 - \max_{\mathbf{z}} \ \mathbb{E}_{\mathbf{Y}|\mathbf{X}} L_\ell(\mathbf{Y}, \mathbf{z}). \end{aligned}$$

□

In general, it is not desired to maximize the risk, but it is shown that both the risk minimizer and maximizer are closely related in the following proposition.

Proposition 3.2. *For any distribution of* $\mathbf{Y}$ *given* $\mathbf{x}$ *and any loss function* L *such that* $\sum_{\mathbf{y}\in\mathcal{Y}} L(\mathbf{y}, \hat{\mathbf{y}}) = c, \forall \hat{\mathbf{y}} \in \mathcal{Y}$*, and* c *is only dependent on the number of labels, the minimum risk is given by*

$$\min_{\mathbf{z}} \ R_L(\mathbf{z}, \mathbf{x}) = c \ - \max_{\mathbf{z}} \sum_{\mathbf{y}\in\mathcal{Y}} L(\mathbf{y}, \mathbf{z}) \cdot (1 - \mathbf{P}_{\mathbf{x}}(\mathbf{y})) \, .$$

Proof.

$$\begin{aligned} \min_{\mathbf{z}} R_L(\mathbf{z}, \mathbf{x}) &= c - c \ - \max_{\mathbf{z}} \sum_{\mathbf{y}\in\mathcal{Y}} L(\mathbf{y}, \mathbf{z}) \cdot (-\mathbf{P}_{\mathbf{x}}(\mathbf{y})) \\ &= c - \sum_{\mathbf{y}\in\mathcal{Y}} L(\mathbf{y}, \mathbf{z}^*) \ - \max_{\mathbf{z}} \sum_{\mathbf{y}\in\mathcal{Y}} L(\mathbf{y}, \mathbf{z}) \cdot (-\mathbf{P}_{\mathbf{x}}(\mathbf{y})) \\ &= c \ - \max_{\mathbf{z}} \sum_{\mathbf{y}\in\mathcal{Y}} L(\mathbf{y}, \mathbf{z}) \cdot (1 - \mathbf{P}_{\mathbf{x}}(\mathbf{y})) \end{aligned}$$

□

Proposition 3.2 is only applicable to coverage if $\sum_{\mathbf{y}\in\mathcal{Y}} L_c(\mathbf{y}, \mathbf{z})$ is a function of n only for any ranking $\mathbf{z}$. This can be easily verified by just noting that any swap of two ranks will result into the same value.

3.2 NP-Completeness of Maximum 2-Search Length

In this section a special case of the decision version of the maximization of the expected search length is proved to be NP-complete. Consequently, the same special case for the decision version of the expected coverage is also NP-complete, as shown in Corollary 3.1. The simplification relies on assuming all instances have exactly two labels which implies that $\mathbf{P}_{\mathbf{x}}(\mathbf{Y})$ has at most $\binom{n}{2}$ non-null values. This special case is denominated as Max 2-SL. We assume that all values of $\mathbf{P}_{\mathbf{x}}(\mathbf{Y})$ are given. It is important to remind that ranking goes from 0 to $n-1$ in this chapter, instead of 1 to n, as discussed in Section 3.1.

For the sake of simplicity, the Max 2-SL problem is reformulated as follows. Consider a weighted complete simple graph $G = (V, E, w)$ with no self-loops, where V is the set of vertices with $n = |V|$, E is the set of edges and w is the symmetric weighting function of the edges such that:

$$0 \leq w(u,v) \leq 1, \forall \{u,v\} \in E \quad \text{and} \sum_{\{u,v\}\in E} w(u,v) = 1. \tag{3.1}$$

Each vertex of G represents a label and, for any $\{u,v\} \in E$, $w(u,v)$ represents the probability of co-occurrence of the labels represented by u and v. Also, consider a bijective function $\phi : V \to \{0, 1, ..., n-1\}$ (representing a ranking function) and an objective function f defined as

$$f_\phi(V,w) = \sum_{\{u,v\}\in E} \min\{\phi(u), \phi(v)\} \cdot w(u,v), \tag{3.2}$$

with $w(v,v) = 0, \ \forall v \in V$. The Max 2-SL problem with a weighted complete graph G can be stated as finding a bijective function $\phi^* : V \to \{0, 1, ..., n-1\}$ maximizing f:

$$f_{\phi^*}(V,w) = \max_{\phi} f_\phi(V,w).$$

Note that function f is composed of only $\mathcal{O}(n^2)$ sums and each sum is composed of basic operations: a multiplication, a minimal of two values and a direct access to a value of ϕ.

The more general case of Max 2-SL where constraints in (3.1) are not imposed is proved to be easily reducible to the original Max 2-SL in Proposition 3.3. For this reason, the constraints in (3.1) are *not* adopted in the rest of this section.

Proposition 3.3. *A modified version of the Max 2-SL in which constraints in* (3.1) *are* not *imposed, can be easily reduced to the original Max 2-SL, that is, an algorithm to solve the original Max 2-SL can be used to solve the Max 2-SL with no constraints in the weights.*

Proof.

Consider the Max 2-SL problem with graph $G = (V, E, w)$. Define a new graph $G' = (V, w')$

with a new weighting function w' as

$$w'(u,v) = a \cdot (w(u,v) - b)$$
$$\text{where} \quad b = \min_{\{x,y\} \in E} w(x,y) \text{ and } \quad a = \sum_{\{x,y\} \in E} (w(x,y) - b)\,.$$

It follows that

$$\max_\phi f_\phi(V, w') = a \cdot \max_\phi f_\phi(V, w) \;-\; b\binom{n}{2}.$$

□

An important form of expressing the function f in (3.2) for a ranking function ϕ is by first defining a set R for each integer $0 \le x < n$ and a real value W for each $v \in V$

$$R_\phi(x) = \{u \in V \mid \phi(u) \ge x\}$$
$$W_\phi(v) = \sum_{u \in R_\phi(\phi(v))} w(u,v),$$

and rewriting f as

$$f_\phi(V,w) = \sum_{v \in V} \phi(v) W_\phi(v). \tag{3.3}$$

Lemma 3.1. *Let $m = |R_\phi(x)|$. For any arbitrary integer $0 \le x < n$ and for any constant value $c \in \mathbb{R}$, if w' is a weighting function where*

$$w'(u,v) = c, \quad \forall u, v \in R_\phi(x),$$

then

$$f_\phi(R_\phi(x), w') = c\binom{m}{3} + cx\binom{m-1}{2}. \tag{3.4}$$

Proof.

See Appendix A.1. □

As a corollary of Lemma 3.1, if the maximum weight value b is positive and the minimum weight value a is negative, then

$$\frac{an^3}{6} \;\le\; f_\phi(U,w) \;\le\; \frac{bn^3}{6}, \text{ for any } U \subseteq V. \tag{3.5}$$

Proof.

Define w' and w'' as two weighting functions where

$$w'(u,v) = a, \quad w''(u,v) = b, \quad \forall u, v \in V.$$

For an arbitrary subset $U \subseteq V$ we have that $f_\phi(V, w') \leq f_\phi(U, w) \leq f_\phi(V, w'')$, which is equivalent to: $f_\phi(R_\phi(0), w') \leq f_\phi(U, w) \leq f_\phi(R_\phi(0), w'')$. Therefore, from Lemma 3.1, it can be shown that

$$\frac{an^3}{6} \leq f_\phi(U, w) \leq \frac{bn^3}{6}.$$

□

For the rest of this section, consider the following definitions.

Definition 3.1. $\mathcal{T}$ is defined as a mapping that transforms any non-weighted graph $G = (V, E)$, to a weighted graph $\mathcal{T}(G) = G' = (V', w)$ such that $V' = V \cup Q$, where $Q = \{q_1, q_2, ..., q_r\}$, $r = 2n^4$, $n = |V|$, $Q \cap V = \varnothing$ and

$$w(u, v) = \begin{cases} 1, & \text{if } \{u, v\} \in E \\ -n, & \text{if } u, v \in V \text{ and } \{u, v\} \notin E \\ 0, & \text{otherwise.} \end{cases}$$

Note that all vertices of Q have edges of zero weight.

Definition 3.2. $\mathcal{P}_\phi(V, Q)$ is defined, for any two disjoint sets of vertices V and Q and any bijective function $\phi : V \cup Q \to \{x \in \mathbb{N}_0 \mid x < |V \cup Q|\}$, as the partition of $V = A_\phi \cup B_\phi \cup C_\phi$ such that

$$\begin{aligned} A_\phi =& \{v \in V \mid \phi(v) < \phi(q), \forall q \in Q\}, \\ B_\phi =& \{v \in V \mid \phi(v) > \phi(q), \forall q \in Q\}, \\ C_\phi =& V - (A_\phi \cup B_\phi). \end{aligned}$$

Therefore $(A_\phi, B_\phi, C_\phi) = \mathcal{P}_\phi(V, Q)$. Note that $\mathcal{P}_\phi(V, Q) \neq \mathcal{P}_\phi(Q, V)$.

The proof of Max 2-SL belonging to NP-HARD relies on reducing the maximum clique problem, proved to belonging to NP-HARD by Karp (1972), into Max 2-SL. The maximum clique problem consists of finding a complete subgraph of a given graph with the highest number of vertices. First, some important lemmas are presented:

Lemma 3.2. *Given a non-weighted graph $G = (V, E)$, let $G' = \mathcal{T}(G) = (V', w)$ and define Q as the set of vertices such that $V' = V \cup Q$ and $V \cap Q = \varnothing$. For an arbitrary solution (ranking) ϕ of Max 2-SL consider the partition $(A_\phi, B_\phi, C_\phi) = \mathcal{P}_\phi(V, Q)$. If there exists a vertex $v \in C_\phi$ in which ($\{u, v\} \in E$, $\forall u \in B_\phi$), then there exists a solution π in which $B_\pi = B_\phi \cup \{v\}$, $C_\pi = C_\phi - \{v\}$ and $f_\pi(V, w) \geq f_\phi(V, w)$.*

Proof.

See Appendix A.2. □

Lemma 3.3. *Given a non-weighted graph $G = (V, E)$, let $G' = \mathcal{T}(G) = (V', w)$ and define Q as the set of vertices such that $V' = V \cup Q$ and $V \cap Q = \varnothing$. For an arbitrary solution (ranking) ϕ of Max 2-SL consider the partition $(A_\phi, B_\phi, C_\phi) = \mathcal{P}_\phi(V, Q)$. If there exists a vertex $v \in (B_\phi \cup C_\phi)$ in which $(\{u, v\} \notin E,\ \exists u \in B_\phi)$, then there exists a solution π in which $A_\pi = A_\phi \cup \{v\}$, $C_\pi = C_\phi - \{v\}$, $B_\pi = B_\phi - \{v\}$ and $f_\pi(V, w) \geq f_\phi(V, w)$.*

Proof.

See Appendix A.3. □

Lemma 3.4. *Given a non-weighted graph $G = (V, E)$, let $G' = \mathcal{T}(G) = (V', w)$, and define Q as the set of vertices such that $V' = V \cup Q$ and $V \cap Q = \varnothing$. For an arbitrary solution (ranking) ϕ of Max 2-SL consider the partition $(A_\phi, B_\phi, C_\phi) = \mathcal{P}_\phi(V, Q)$. There exists a ranking π in which $f_\pi(V, w) \geq f_\phi(V, w)$, $C_\pi = \varnothing$ and B_π is a clique in G.*

Proof.

See Appendix A.4. □

Theorem 3.1. *Maximum clique is reducible to maximum 2-Search Length*

Proof.

Given a graph $G = (V, E)$ and a positive integer k (this integer comes from the decision version) as input for the maximum clique problem, let $n = |V|$. Consider only the cases where $n \geq 2$ and $k \geq 2$. The decision version of maximum clique problem is: **Is there at least one clique of size k in G?**

Now, let be defined the decision version for the maximum 2-search length. It consists of a weighted complete graph $G' = (V', w)$, $n = |V'|$ and a positive integer Z (this integer comes from the decision version) defined as

$$Z = n^4 k(k-1) - n^4.$$

A bijective function ϕ^* is called a satisfying solution of Max 2-SL if $f_{\phi^*}(V', w) \geq Z$. The decision version of maximum 2-search length is: **Is there at least one satisfying solution in G'?**

What it needs to be shown is that one problem can be solved by the other (efficiently). The transformation from one to another is given by $G' = \mathcal{T}(G)$. Before proceeding, consider the following definitions. Let r be defined as $r = 2n^4$. Define Q as the set of vertices such that $V' = V \cup Q,\ V \cap Q = \varnothing$. Consider the partition $(A_\phi, B_\phi, C_\phi) = \mathcal{P}_\phi(V, Q)$. Note that, from (3.3), $f_\phi(V', w)$ can be reformulated as

$$\begin{aligned} f_\phi(V', w) = \sum_{v \in A_\phi} \phi(v) W_\phi(v) \ &+ \ \sum_{v \in B_\phi} \phi(v) W_\phi(v) \\ &+ \ \sum_{v \in C_\phi} \phi(v) W_\phi(v). \end{aligned} \tag{3.6}$$

Let us first suppose there exists a satisfying solution ϕ and let $l = |Q| + |A_\phi| = 2n^4 + |A_\phi|$ and $x = |B_\phi|$. To show that there exists a clique of size of at least k, assume, from Lemma 3.4, that $C_\phi = \varnothing$ and $w(u, v) = 1$ for all $u, v \in B_\phi$, so, from (3.6), it follows that

$$f_\phi(V', w) = f_\phi(A_\phi, w) \ + \ f_\phi(B_\phi, w).$$

Considering that $B_\phi = R_\phi(l)$ and from Lemma 3.1, it can be concluded that:

$$f_\phi(V', w) = f_\phi(A_\phi, w) \ + \ \binom{x}{3} + l\binom{x}{2}.$$

By using the inequality (3.5) it can be concluded that

$$\begin{aligned} n^4 k(k-1) - n^4 &\le f_\phi(V', w) \\ &= f_\phi(A_\phi, w) \ + \ \binom{x}{3} + (2n^4 + |A_\phi|)\binom{x}{2} \\ &\le \frac{n^3}{6} + \binom{x}{3} + (2n^4 + n)\binom{x}{2} \\ &\le \frac{n^3}{6} + \frac{n^3}{6} + 2n^4\binom{x}{2} + \frac{n^3}{2} \\ &= n^3\left(nx(x-1) + \frac{5}{6}\right) \ \le \ n^4\left(x(x-1) + \frac{5}{6}\right). \end{aligned}$$

By dividing both sides by n^4, it can be concluded that

$$k(k-1) - 1 \le x(x-1) + \frac{5}{6}.$$

The above inequality does not hold for $x \le k-1$. Therefore, $x \ge k$. This implies that B_ϕ is a clique in G with $|B_\phi| \ge k$.

Now, suppose there exists a clique $C \subseteq V$, with $C = \{c_1, c_2, ..., c_{|C|}\}$ where $|C| \ge k$ and let $\bar{C} = \{\bar{c}_1, \bar{c}_2, ..., \bar{c}_{|\bar{C}|}\}$ be $\bar{C} = V - C$. To prove that there exists a satisfying solution of Max 2-SL for graph G', define $\phi : V' \to \mathbb{N}$ as follows:

$$\begin{aligned} \phi(\bar{c}_i) =& i - 1, && 1 \le i \le |\bar{C}|; \\ \phi(q_i) =& i - 1 + |\bar{C}|, && 1 \le i \le |Q|; \\ \phi(c_i) =& i - 1 + |\bar{C}| + |Q|, && 1 \le i \le |C|. \end{aligned}$$

Let d be defined as

$$d \;=\; f_\phi(V', w) - Z \;=\; f_\phi(\bar{C}, w) + f_\phi(C, w) - Z.$$

Let $l = |\bar{C}| + |Q|$. Considering that $C = R_\phi(l)$ and from Lemma 3.1, it can be concluded that:

$$f_\phi(C, w) = \binom{k}{3} + l\binom{k}{2} \;\geq\; n^4 k(k-1). \tag{3.7}$$

The integer d is shown to be positive by taking (3.5) and (3.7), so

$$\begin{aligned} d &= f_\phi(\bar{C}, w) + f_\phi(C, w) - n^4 k(k-1) + n^4 \\ &\geq f_\phi(\bar{C}, w) + n^4 k(k-1) - n^4 k(k-1) + n^4 \\ &\geq \frac{-n^4}{6} + n^4. \end{aligned}$$

This completes the proof of the theorem. □

3.3 Two correlated Coverage

In the previous section, it has been proved that the optimal expected coverage is NP-HARD even for a label cardinality of 2 for all instances. In this section, it will be shown that the optimal expected coverage is still NP-HARD even when assuming low level of label dependencies. Therefore, it is crucial to define what is a low level of label dependency. Before that, consider the following definitions:

Definition 3.3. For any labelling $\mathbf{Y} = (Y_1, \ldots, Y_n)$ of size n and for any subset $A \subseteq \{1..n\}$, let $\mathbf{Y}_A$ be defined as the "sub-labelling" of labels in set A. Example: for $A = \{1, 3\}$, it has that $\mathbf{Y}_A = (Y_1, Y_3)$. Since a set does not have a specific order, we consider that the vector $\mathbf{Y}_A$ is always in order, i.e, $Y_{\{4,2\}} = (Y_2, Y_4)$.

Definition 3.4. For any labelling $\mathbf{Y} = (Y_1, \ldots, Y_n)$ and for any ranking $\mathbf{z}$ of n labels, let $Y_i^{(\mathbf{z})}$, for $1 \leq i \leq n$, be defined as the label ranked at i-th position by $\mathbf{z}$. This means that $Y_1^{(\mathbf{z})}$ is the first ranked label and $Y_n^{(\mathbf{z})}$ is the last ranked label. For instance, for $\mathbf{z} = (2, 3, 1, 4)$, it has that $Y_1^{(\mathbf{z})} = Y_3$ and $Y_2^{(\mathbf{z})} = Y_1$.

Definition 3.5. For any ranking $\mathbf{z}$, let $T_k^{\mathbf{z}}$ be defined as the set of all label indices ranked by $\mathbf{z}$ with a rank greater than or equal to k.

For defining the level of dependence of labels, consider the parametrization given by Teugels (1990), where any arbitrary multivariate Bernoulli distribution of n variables $\mathbf{Y} = (Y_1, ..., Y_n)$ can be represented by these $2^n - 1$ parameters:

$$p_i = 1 - q_i = \mathbb{E}\left[Y_i\right] = \mathbf{P}(Y_i = 1), \qquad \text{The marginal distribution,}$$

and all first-order central moments of all combinations of variables:

$$\begin{aligned}
\delta_{ij} =& \mathbb{E}\left[(Y_i - p_i)(Y_j - p_j)\right], && \text{for all distinct pairs } (Y_i, Y_j),\\
\delta_{ijk} =& \mathbb{E}\left[(Y_i - p_i)(Y_j - p_j)(Y_k - p_k)\right], && \text{for all distinct } (Y_i, Y_j, Y_k),\\
& \ldots \\
\delta_{1..n} =& \mathbb{E}\left[(Y_1 - p_1)(Y_2 - p_3)\cdots(Y_n - p_n)\right].
\end{aligned}$$

Then, $\mathbf{P}(\mathbf{Y})$ can be represented as a polynomial of these $2^n - 1$ parameters. An example for $n = 3$ and $\mathbf{Y} = (1, 1, 1)$:

$$\mathbf{P}(1, 1, 1) = p_1p_2p_3 + p_3\delta_{12} + p_2\delta_{13} + p_1\delta_{23} + \delta_{123}.$$

The δ_{ij} represents the two-way dependence among Y_i and Y_j, while δ_{ijk} represents the three-way dependence among Y_i,Y_j and Y_k. It is important to note that δ_{ij} is the covariance of Y_i and Y_j, and the covariance of two Bernoulli variables are zero if and only if they are independent of each other. For three or more variable, δ is only zero iff at least one of the variables is independent of all others. In this sense, the δs are called dependence parameters.

Since we like to study the optimal expected coverage when there are only low dependencies among labels, its seems reasonable to adopt the scenario where the δ for three or more variables are all zero. Consequently, the joint distribution can be represented with only a quadratic number of parameters, instead of $2^n - 1$. We call this scenario as two-correlated MultiVariate Bernoulli distribution (2-MVB). Let

$$\mathbf{M}(\mathbf{y}) = \prod_{k=1}^{n} p_k^{y_k} q_k^{1-y_k},$$

then the general form of $\mathbf{P}(\mathbf{Y} = \mathbf{y})$ for 2-MVB becomes (TEUGELS, 1990):

$$\mathbf{P}(\mathbf{Y} = \mathbf{y}) = \mathbf{M}(\mathbf{y}) + \sum_{(i,j):j>i} \left(-1^{y_i+y_j} \cdot \delta_{ij} \frac{\mathbf{M}(\mathbf{y})}{p_i^{y_i} q_i^{1-y_i} p_j^{y_j} q_j^{1-y_j}} \right), \tag{3.8}$$

where $p_i = 1 - q_i$. A particular case of interest is when a subset of variables/labels are zeroes (why this case is interesting is shown at Proposition 3.4): Let $A \subseteq \{1..n\}$ be this arbitrary subset of variables indices, then

$$\mathbf{P}(\mathbf{Y}_A = \mathbf{0}_{|A|}) = \prod_{i \in A} q_i \ + \sum_{(i,j)\in A:j>i} \delta_{ij} \frac{\prod_{i\in A} q_i}{q_i q_j}, \qquad \text{See Definition 3.3} \tag{3.9}$$

where $\mathbf{0}_{|A|}$ is the vector of $|A|$ zeroes. Example for $A = \{1, 2, 3\}$:

$$\mathbf{P}(\mathbf{Y}_{\{1,2,3\}} = (0, 0, 0)) = q_1q_2q_3 + q_3\delta_{12} + q_2\delta_{13} + q_1\delta_{23}.$$

Equation (3.9) can be further simplified, because the findings presented in this section assume that $p_1 = p_2 = \cdots = p_n = p = 1 - q$:

$$\mathbf{P}(\mathbf{Y}_A = \mathbf{0}_{|A|}) = q^{|A|} \ + \ q^{|A|-2} \sum_{(i,j)\in A:j>i} \delta_{ij}. \tag{3.10}$$

Now, it will be shown in Proposition 3.4 that the expected coverage can be easily calculated for a particular ranking $\mathbf{z}$ using only n parameters of the label distribution. Also, it will be shown why $\mathbf{P}(Y_A = \mathbf{0}_{|A|})$ (Equation 3.10) is important for analysing the expected coverage.

Proposition 3.4. *For any distribution* $\mathbf{P}$ *of* $\mathbf{Y}$*, and for any ranking* $\mathbf{z}$ *of* n *labels, and considering Definition 3.3 and 3.4, it has that*

$$\begin{aligned}\mathbb{E}_{\mathbf{Y}}\left[L_c(\mathbf{Y},\mathbf{z})\right] &= n - \mathbf{P}(Y_n^{(\mathbf{z})} = 0) - \mathbf{P}(Y_n^{(\mathbf{z})} = 0, Y_{n-1}^{(\mathbf{z})} = 0) - \mathbf{P}(Y_n^{(\mathbf{z})} = 0, Y_{n-1}^{(\mathbf{z})}, Y_{n-2}^{(\mathbf{z})} = 0) - \dots \\ &= n - \sum_{i=1}^{n} \mathbf{P}(\mathbf{Y}_{\{i..n\}}^{(\mathbf{z})} = \mathbf{0}).\end{aligned}$$

Proof.

This comes easily as a consequence of the relationship between the expected value of a random variable and its cumulative density function:

$$\mathbb{E}\left[L_c\right] = \sum_{i=0}^{n-1}(1 - \mathbf{P}(L_c \leq i)) = n - \sum_{i=0}^{n-1}\mathbf{P}(L_c \leq i),$$

where the loss L_c is considered a random variable, since it depends on $\mathbf{Y}$. Keep in mind that coverage can only have values in $\{0, 1, ..., n\}$. To answer when $L_c \leq i$ occurs, it is easier to answer when $L_c > i$ not occurs, although they are equivalent statements, i.e $\mathbf{P}(L_c \leq i) = 1 - \mathbf{P}(L_c > i)$. By definition of coverage, $L_c > i$ does not occur if and only if all last ranked labels until rank $i + 1$ are zeroes, i.e, $Y_{i+1}^{(\mathbf{z})} = Y_{i+2}^{(\mathbf{z})} = \cdots = Y_n^{(\mathbf{z})} = 0$. Hence, $\mathbf{P}(L_c \leq i) = \mathbf{P}(\mathbf{Y}_{\{i+1,\dots,n\}}^{(\mathbf{z})} = \mathbf{0})$, and consequently

$$\mathbb{E}\left[L_c\right] = n - \sum_{i=0}^{n-1}\mathbf{P}(\mathbf{Y}_{\{i+1,\dots,n\}}^{(\mathbf{z})} = \mathbf{0}).$$

□

Finally, it will be proved that 2-MVB problem is NP-HARD.

Theorem 3.2. *Optimal 2-MVB is NP-HARD, by a reduction of the maximum clique problem.*

Proof.

Given a graph $G = (V, E)$ for the maximum clique problem, the corresponding input for maximum two correlated expected coverage is a multi-label problem of $n = |V|$ labels with the probability distribution $\mathbf{P}$ of $\mathbf{Y} = (Y_1, ..., Y_n)$ where

$$\begin{aligned}
&\mathbf{P}(Y_1 = 1) = \mathbf{P}(Y_2 = 1) = \cdots = p = 1 - q = \frac{1}{n^3}, \\
&\mathrm{Cov}\left[Y_i, Y_j\right] = \delta_{ij} = \begin{cases} 0, & \text{if } \{i,j\} \notin E, \\ n^{-8}, & \text{if } \{i,j\} \in E, \end{cases} \qquad (3.11) \\
&\forall A \subseteq \{1..n\},\ \text{such that } |A| \geq 3, \text{we have that } \mathbb{E}\left[\prod_{i \in A}(Y_i - p)\right] = 0.
\end{aligned}$$

Note that, each label represents a vertex. It will be proved that such a distribution actually exists, i.e, $\mathbf{P}(\mathbf{Y} = \mathbf{y}) \geq 0$ for all labelling $\mathbf{y}$.

$$\begin{aligned}
\frac{\mathbf{P}(\mathbf{Y} = \mathbf{y})}{\mathbf{M}(\mathbf{y})} &= 1 + \sum_{i,j:j>i}\left(\underbrace{-1^{y_i+y_j}}_{\geq -1} \cdot \delta_{ij} \frac{1}{p^{y_i} q^{1-y_i} p^{y_j} q^{1-y_j}}\right) && \text{From (3.8)} \\
&\geq 1 - \sum_{i,j:j>i} \underbrace{\delta_{ij}}_{\leq n^{-8}} \frac{1}{p^{y_i} q^{1-y_i} p^{y_j} q^{1-y_j}} \\
&\geq 1 - \sum_{i,j:j>i} \frac{n^{-8}}{q^2} = 1 - \sum_{i,j:j>i} \frac{n^{-8}}{n^{-6}} \\
&= 1 - \sum_{i,j:j>i} n^{-2} = 1 - \binom{n}{2} \cdot n^{-2} \\
&\geq 1 - \frac{1}{2} > 0.
\end{aligned}$$

It will be proved that if there exists a clique of size c in G, then the c lowest ranked labels of the optimal ranking for coverage will represent a clique (the transformation has a one-to-one relation of labels and vertices). From Proposition 3.4, it has that

$$\begin{aligned}
\mathbb{E}\left[L_c\right] &= \sum_{i=0}^{n-1}(1 - \mathbf{P}(L_c \leq i)) - n = -\sum_{i=1}^{n} \mathbf{P}(\mathbf{Y}^{(\mathbf{z})}_{\{i..n\}} = \mathbf{0}) \\
&= -\underbrace{\mathbf{P}(Y^{(\mathbf{z})}_n = 0)}_{q} - \sum_{i=2}^{n} \mathbf{P}(\mathbf{Y}^{(\mathbf{z})}_{\{i..n\}} = \mathbf{0}) \\
&= -q - \left(\sum_{i=2}^{n} q^i + q^{i-2} \sum_{j,k \in T^{\mathbf{z}}_i : k>j} \delta_{jk}\right) && \text{From (3.10) and Definition 3.5} \\
&= -\sum_{i=1}^{n} q^i \ - \left(\sum_{i=2}^{n} q^{i-2} \sum_{j,k \in T^{\mathbf{z}}_i : k>j} \delta_{jk}\right),
\end{aligned}$$

where $j, k \in T^{\mathbf{z}}_i : k > j$ means all distinct pairs of label indices ranked with a rank equal or above i. For the sake of simplicity, let $\delta_{jj} = 0$ for any j and let us use the fact that $\delta_{jk} = \delta_{kj}$:

$$-\sum_{i=1}^{n} q^i \ - \left(\sum_{i=2}^{n} q^{i-2} \sum_{j,k \in T^{\mathbf{z}}_i : k>j} \delta_{jk}\right) = -\sum_{i=1}^{n} q^i \ - \left(\sum_{i=2}^{n} \frac{q^{i-2}}{2} \sum_{j,k \in T^{\mathbf{z}}_i} \delta_{jk}\right)$$

As we are studying the optimization and the NP-HARDNESS of this above equation/problem, the constant term $-\sum_{i=1}^{n} q^i$ is negligible and does not affect the solution/hardness of the problem. Hence, the objective function will be changed to remove the constant term and multiplied by -2 to simplify equations:

$$\begin{aligned}\operatorname*{argmin}_{\mathbf{z}} \mathbb{E}\left[L_c\right] &= \operatorname*{argmax}_{\mathbf{z}} -2\ \mathbb{E}\left[L_c\right] \\ &= \operatorname*{argmax}_{\mathbf{z}} \left(\sum_{i=2}^{n} q^{i-2} \sum_{j,k \in T_i^{\mathbf{z}}} \delta_{jk} \right).\end{aligned}$$

Note that at each succession of iteration of i, the sum of deltas $\sum_{j,k} \delta_{jk}$ contributes less to the total sum, since it is multiplied by q^{i-1} instead of q^{i-2}. The idea of choosing a very low value for q is to make the sum $q^{i-1} \sum_{j,k \in T_{i+1}^{\mathbf{z}}} \delta_{jk}$ irrelevant compared to $q^{i-2} \sum_{j,k \in T_i^{\mathbf{z}}} \delta_{jk}$. This will be proved formally now, by contradiction: for any i if there exists $j, k \in T_i^{\mathbf{z}}$ such that $\delta_{jk} = n^{-8}$ (i.e there exists $\{j,k\} \in E$) then

$$\begin{aligned} \sum_{\ell \geq i+1}^{n} \underbrace{q^{\ell-2}}_{\leq q^{i-1}} \sum_{j,k \in T_\ell^{\mathbf{z}}} \delta_{jk} &> q^{i-2} \sum_{j,k \in T_i^{\mathbf{z}}} \delta_{jk} && \text{(Assumption)} \\ q^{i-1} \sum_{\ell > i}^{n} \sum_{j,k \in T_\ell^{\mathbf{z}}} \delta_{jk} &> q^{i-2} \sum_{j,k \in T_i^{\mathbf{z}}} \delta_{jk} \\ q \sum_{\ell > i}^{n} \sum_{j,k \in T_\ell^{\mathbf{z}}} \underbrace{\delta_{jk}}_{\leq n^{-8}} &> \underbrace{\sum_{j,k \in T_i^{\mathbf{z}}} \delta_{jk}}_{\geq n^{-8}} \\ \underbrace{q}_{n^{-3}} \underbrace{\sum_{\ell > i}^{n} \sum_{j,k \in T_\ell^{\mathbf{z}}} n^{-8}}_{\text{less than } n^3 \text{ terms}} &> n^{-8} \\ n^{-3} \cdot n^{3} \cdot n^{-8} &> n^{-8}. && \text{(Contradiction!)} \end{aligned}$$

Hence, it is true that for any $\{j,k\} \in E$ and $1 \leq i < n$, then

$$\sum_{\ell \geq i+1}^{n} q^{\ell-2} \sum_{j,k \in T_\ell^{\mathbf{z}}} \delta_{jk} \leq q^{i-2} \sum_{j,k \in T_i^{\mathbf{z}}} \delta_{jk}.$$

With this proved, it is clear that placing a clique in the best/lowest ranks is a necessary condition for optimizing the 2-MVB problem. □

To determine if the optimal two correlated coverage problem is in NP, we can look at the equation in Proposition 3.4, which gives a way to calculate the expected coverage for any particular ranking. Since each term of the sum $\sum_{i=1}^{n} \mathbf{P}(\mathbf{Y}_{\{i..n\}}^{(\mathbf{z})} = \mathbf{0})$ can be calculated in polynomial time using Equation 3.10, the optimal two correlated coverage is in NP. Hence, the optimal two correlated coverage is NP-COMPLETE. If one is concerned with the scenario where all labels are independent, coverage can be computed in $\mathcal{O}(n \log n)$ by just sorting labels according to their marginal probability.

Chapter Conclusion

It was shown that computing the minimal minimum expected coverage is identical with respect to the complexity of computing the maximal minimum expected search length. It was also proved that a specific scenario of risk minimization of coverage and search length belongs to the class NP-COMPLETE. Moreover, even assuming a low level of dependence among labels, the optimal expected coverage is still in the classNP-COMPLETE. Therefore, as expected, the general version belongs to the class NP-HARD, which shows how much the computation of the optimal solution in the general version is intractable hard. This result suggests that algorithm designers should be stimulated to work on other more promising research topics such as finding an approximate algorithm for the problem. This leads to a clear new future work: finding efficient approximated algorithms for the problem. Another interesting future work is analysing the sample complexity of learning the coverage loss function, that is, the necessary number of training samples in order to achieve a coverage below a desired value.

4 Regret Analysis of Calibrated Label Ranking

In this chapter, the predictions of ranking by pairwise classification and its extension for classification, calibrated label ranking, are analyzed in terms of regret. In this sense, mathematical proofs are given showing its performance on the worst case scenario for five multi-label metrics.

Empirical evidence clearly shows RPC being good at optimizing metrics that take rank into account, while also showing that CLR is not competitive against state-of-the-art methods on example-based metrics such as F-measure (TROHIDIS et al., 2011; ZHANG; SCHNEIDER, 2012a; ZHANG; SCHNEIDER, 2012b; WANG et al., 2014; TAHIR; KITTLER; BOURIDANE, 2016; HUANG et al., 2019). Although these results show at average where (i.e. which metric) does CLR is good/bad, they do not show why CLR or RPC are good/bad and neither when (i.e. which type of dataset).

The main motivation for the work presented in this chapter comes from the problem of not knowing exactly why and when both CLR and RPC is good/bad in specific situations for specifics metrics. Such explanations help researchers choose and better understand their multi-label methods. Therefore, the main objective in this chapter is provide an explanation for the CLR and RPC performance.

This chapter presents interesting theoretical properties of CLR and RPC that shows when it should not be used. It shows a major issue in the way they make its pairwise comparison, resulting in poor performance for a very particular probability label distribution type. As other authors already suggested, the issue lies mainly on how the probability $\mathbf{P}(Y_i = 1 | Y_i = 1 \text{ xor } Y_j = 1)$ is used inside the prediction. The results suggest CLR should be taken with caution when $\mathbf{P}(Y_i = 1 \oplus Y_j = 1)$ is close to zero for some labels i and j.

Section 4.1 presents the worst case analysis for RPC and CLR on five multi-label metrics on two scenarios: one general scenario with high label dependencies and a specific scenario of low label dependencies. In Section 4.2, a note is made regarding the relationship between the average case scenario and the worst case scenario. The chapter ends with conclusions about the performance of RPC and CLR.

4.1 Worst case analysis of Calibrated Label Ranking

In this section, some probability distributions are constructed to show the performance of CLR and RPC in the worst case scenario. One of these special distributions where is defined as following. Denote 0_n as a vector of n zeroes, 1_n as a vector of n ones and $\mathbf{y}^{(i)}$ as a n-dimensional vector of zeros apart from a one at the i-th position. Let $\hat{\mathbf{P}}_m$ denote a special distribution of $\mathbf{Y}$ such that

$$\hat{\mathbf{P}}_m(\mathbf{y}) = \begin{cases} \frac{m+1}{2(n+1)}, & \text{if } \mathbf{y} = 1_n \\ \epsilon, & \text{if } \mathbf{y} = \mathbf{y}^{(i)} \text{ for any } 1 \leq i \leq m \\ 1 - \frac{m+1}{2(n+1)} - \epsilon \cdot m, & \text{if } \mathbf{y} = 0_n \\ 0, & \text{otherwise} \end{cases}$$

where m is a positive integer such that $0 < m < n$ and ϵ is an arbitrary positive real number that is assumed to be “really close” to 0. An example of $\hat{\mathbf{P}}_2$ for $n = 4$:

$$\begin{aligned} \hat{\mathbf{P}}_2(0,0,0,0) &= 70\% - 2\epsilon \\ \hat{\mathbf{P}}_2(1,1,1,1) &= 30\% \\ \hat{\mathbf{P}}_2(1,0,0,0) &= \hat{\mathbf{P}}_2(0,1,0,0) = \epsilon, \end{aligned}$$

where null probabilities are omitted. The most important point to note about $\hat{\mathbf{P}}_m$ is the high probability of occurrence of labelling 0_n, specially when m is low. Also, note that $\hat{\mathbf{P}}_m$ has exactly $m + 2$ non-null values. The purpose of ϵ is to avoid undefined values when calculating f (e.g $\frac{0}{0}$) and to conveniently manipulate the output of function f.

Proposition 4.1 shows an important property of this distribution.

Proposition 4.1. *When considering distribution $\hat{\mathbf{P}}_m$, CLR predicts ones for the first m labels and zeroes for the other labels, i.e, $\sum_{i=1}^{m} h_i^{clr} = \sum_{i=1}^{n} h_i^{clr} = m$.*

Proof.

See Appendix B.1. □

This proposition shows how much CLR is sensible to conditional probabilities. Just an arbitrarily small value ϵ in $\hat{\mathbf{P}}_m$ makes CLR predicts m labels incorrectly. Next, it will be seen how much this impacts CLR performance where several theorems with respect to the regret of CLR and RPC are presented. Following each theorem, relevant observations are made.

Theorem 4.1. *The following upper bound holds for the regret with respect to Hamming loss:*

$$\sup_{\mathbf{P} \in \mathcal{P}_n} (r_H(\mathbf{h}^{clr})) = \begin{cases} \frac{n}{4(n+1)}, & \textit{if } n \textit{ is even} \\ \frac{n-1}{4n}, & \textit{if } n \textit{ is odd,} \end{cases}$$

where $\mathcal{P}_n$ denotes the set of all distributions over n labels such that $\mathbf{P}^{(i)} \leq \frac{1}{2}$ for all i.

Proof.

See Appendix B.2. □

An interesting point to observe from Theorem 4.1 is that there exists at least one distribution in the family $\mathcal{P}_n$ such that $r_H(\mathbf{h}^{\text{clr}}) \leq \frac{1}{4}$. Empirically, CLR and BR (a.k.a one-against-all) has been shown to have a much closer performance on average with respect to Hamming loss, according to experiments in the literature (FÜRNKRANZ et al., 2008; TROHIDIS et al., 2011; ZHANG; SCHNEIDER, 2012a; ZHANG; SCHNEIDER, 2012b; WANG et al., 2014).

A more interesting result is presented with respect to subset 0/1 loss in Theorem 4.2.

Theorem 4.2. *The following lower bound holds for the regret with respect to subset 0/1 loss:*

$$\sup_{\mathbf{P}} r_s(\mathbf{h}^{clr}) \geq \frac{n}{n+1}.$$

Proof.

Consider the regret $r_s(\mathbf{h}^{\text{clr}})$ on distribution $\hat{\mathbf{P}}_m$ for $m = 1$. If ϵ is sufficiently small, then the mode of $\hat{\mathbf{P}}_1$ is 0_n, and $\hat{\mathbf{P}}_1(0_n) = 1 - \frac{1}{n+1} - \epsilon$. From Proposition 4.1, it has that $\hat{\mathbf{P}}_1(\mathbf{h}^{\text{clr}}) = \hat{\mathbf{P}}_1(\mathbf{y}^{(1)}) = \epsilon$. As the mode of distribution is an optimal labelling for subset 0/1 loss, the regret on distribution $\hat{\mathbf{P}}_m$ can be written as

$$\begin{aligned} r_s(\mathbf{h}^{\text{clr}}) &= \hat{\mathbf{P}}_1(0_n) - \hat{\mathbf{P}}_1(\mathbf{y}^{(1)}) \\ &= 1 - \frac{1}{n+1} - 2\epsilon \end{aligned}$$

The value of ϵ can be arbitrarily small, so the supremum of $r_s(\mathbf{h}^{\text{clr}})$ is at least $1 - \frac{1}{n+1} = \frac{n}{n+1}$. □

Theorem 4.2 shows that when n tends to infinity, the supremum of regret $r_s(\mathbf{h}^{\text{clr}})$ tends to 1, which is the highest regret possible for subset 0/1 loss. A high regret is already expected as seen in empirical evidence (TROHIDIS et al., 2011; ZHANG; SCHNEIDER, 2012a; ZHANG; SCHNEIDER, 2012b; WANG et al., 2014; TAHIR; KITTLER; BOURIDANE, 2016; HUANG et al., 2019; SUN; GE; KANG, 2019), but surely not of such magnitude. It is important to note that even for a small number of labels, the worst case regret is high, e.g. for $n = 4$ the highest regret is at least 0.8.

Theorem 4.3. *The following lower bound holds for the regret with respect to Jaccard distance:*

$$\sup_{\mathbf{P}} r_J(\mathbf{h}^{clr}) \geq 1 - \frac{1}{n}.$$

Proof.

See Appendix B.3. □

Note that if $n \to \infty$, then $r_J(\mathbf{h}^{\text{clr}})$ tends to 1. Again, this is the highest regret possible for Jaccard distance and the regret is also high even for small n, e.g. for $n = 4$ the highest regret is at least 0.75. A high regret was already expected, but not this high. This is another metric researchers should be aware when considering the worst case.

Theorem 4.4. *The following lower bound holds for the regret with respect to F-measure:*

$$\sup_{\mathbf{P}} r_F(\mathbf{h}^{clr}) \geq 1 - \frac{n+3}{(n+1)^2}.$$

Proof.

See Appendix B.4. □

Note that if $n \to \infty$, then $r_F(\mathbf{h}^{\text{clr}})$ tends to 1, which is the highest possible regret for F-measure. For small values of n, the highest regret is still high, e.g. for $n = 4$ the highest regret is at least 0.72.

Another interesting result is shown for rank loss in Theorem 4.5, where RPC does not achieve optimal regret.

Theorem 4.5. *For any n divisible by 4, the following lower bound holds for the regret with respect to normalized rank loss:*

$$\sup_{\mathbf{P}} r_{\hat{r}}(\mathbf{h}^{rpc}) \geq \frac{1}{6}.$$

Proof.

See Appendix B.5. □

Although Theorem 4.5 is not conclusive for stating that RPC performs poorly at worst case scenarios, it suggests that RPC does not optimize rank loss for $n \geq 4$, which is not the expected behaviour. The non-optimal performance for RPC does not occur for the same reason as the CLR: the function f can be 1 even when the label cardinality is very low.

As it can be seen in the proofs, the poor performance of CLR in the worst case scenario comes from giving too much importance to specific conditional probabilities:. For instance, an arbitrarily small value ϵ is enough to change the conditional probability at f from zero to one and consequently changing classification. The expected value of multi-label metrics does not give such importance to conditional probabilities, as it can be seen in their formulas at equations (2.13), (2.14) and others presented by Dembczyński et al. (2012).

It is natural to question how rare are the special distributions used in this work in this section and if there are other distributions that yield similar results. Moreover, it is already expected that CLR achieves a non-optimal performance on distributions that yields more than pairwise dependencies, since CLR is specifically designed to exploit dependencies among pairs of labels. Despite this, it will be shown that CLR can achieve a poor performance on F-measure and subset 0/1 loss even when considering a distribution with only pairwise dependencies. For this purpose, let us define a family of probability distributions $\mathcal{P}$ of n labels such that for any $\bar{\mathbf{P}} \in \mathcal{P}$:

$$\bar{\mathbf{P}}(\mathbf{y}) = \bar{\mathbf{P}}(y_1, y_2) \cdot \bar{\mathbf{P}}(y_3) \cdot \bar{\mathbf{P}}(y_4) \;\cdots\; \bar{\mathbf{P}}(y_n) = \bar{\mathbf{P}}(y_1, y_2) \cdot \prod_{i=3}^{n} \bar{\mathbf{P}}(y_i),$$

where the probabilities $\bar{\mathbf{P}}(y_1, y_2)$ and $\bar{\mathbf{P}}(y_i)$ are abbreviations of $\bar{\mathbf{P}}(Y_1 = y_1, Y_2 = y_2)$ and $\bar{\mathbf{P}}(Y_i = y_i)$, respectively. Any $\bar{\mathbf{P}} \in \mathcal{P}$ is constructed such that it can be written as a function of only the joint distribution of two labels and the marginal distributions of the other labels. It is "almost" a distribution of independent variables. Not all probability distributions can be written in this form, because the joint distribution of three or more labels cannot be decomposed generically to the joint probability of only one or two labels. Readers are recommended to check the work of Teugels (1990), if interested in more details about decomposing and understanding the joint probability of a multivariate Bernoulli distribution. In order to show some properties of CLR, let a specific distribution $\bar{\mathbf{P}} \in \mathcal{P}$ be defined such that

$$\bar{\mathbf{P}}(y_1, y_2) = \begin{cases} 3\epsilon, & \text{if } y_1 = y_2 = 0, \\ \epsilon, & \text{if } y_1 = y_2 = 1, \\ \frac{1}{2} - 2\epsilon, & \text{if } y_1 \neq y_2, \end{cases}$$

and $\bar{\mathbf{P}}(Y_i = 1) = \phi_n$ for $i \geq 3$, where ϕ_n is a function of n such that:

$$0 \leq \phi_n < \frac{\epsilon}{3n} \quad \text{and} \quad \lim_{n\to\infty} (1 - \phi_n)^n = 1,$$

for all $n \geq 3$. There are many functions satisfying these two conditions of ϕ_n, for instance, $\phi_n = \epsilon/n^2$. It is crucial to note that if $\epsilon \approx 0$, then $\phi_n \approx 0$ and, consequently, $\bar{\mathbf{P}}$ will have only two labellings ($\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)}$) being with significant probabilities. Indeed,

$$\bar{\mathbf{P}}(\mathbf{y}^{(1)}) + \bar{\mathbf{P}}(\mathbf{y}^{(2)}) = (1 - 4\epsilon) \cdot (1 - \phi_n)^{(n-2)},$$

which tends to 1 as ϵ goes to 0. Before stating about the regret of CLR on distribution $\bar{\mathbf{P}}$, it is important to take a look at a property of $\bar{\mathbf{P}}$ stated in Proposition 4.2.

Proposition 4.2. *For distribution $\bar{\mathbf{P}}$ of n labels, CLR will predict 0_n.*

Proof.

See Appendix B.6. □

Theorem 4.6 shows the regret of CLR with respect to subset 0/1 loss in $\bar{\mathbf{P}}$.

Theorem 4.6. *The following expression holds for the regret with respect to subset 0/1 loss*

$$r_s(\mathbf{h}^{clr}) = \left(\frac{1}{2} - 5\epsilon\right) \cdot (1 - \phi_n)^{n-2}, \text{ for distribution } \bar{\mathbf{P}},$$

and, consequently

$$\lim_{n\to\infty,\epsilon\to 0} r_s(\mathbf{h}^{clr}) = \frac{1}{2}.$$

Proof.

See Appendix B.7. □

One half is a much better regret than 1, but is surely high considering such a simple distribution as $\bar{\mathbf{P}}$. This is further evidence that it is not enough to exploit dependencies to improve performance, even considering only pairwise dependencies. The same argument can be used for the regret with respect to Jaccard distance, as shown in Theorem 4.7.

Theorem 4.7. *The following expression holds for the regret with respect to Jaccard distance*

$$\lim_{\epsilon\to 0} r_J(\mathbf{h}^{clr}) = \frac{1}{2}, \text{ for distribution } \bar{\mathbf{P}}.$$

Proof.

See Appendix B.8. □

Theorem 4.8 shows the regret of CLR with respect to F-measure in $\bar{\mathbf{P}}$.

Theorem 4.8. *The following expression holds for the regret with respect to F-measure loss*

$$\lim_{\epsilon\to 0} r_J(\mathbf{h}^{clr}) = \frac{2}{3}, \text{ for distribution } \bar{\mathbf{P}}.$$

Proof.

See Appendix B.9. □

Similar to Theorem 4.6, CLR achieves a poor performance in such a simple distribution, therefore it is not enough to exploit dependencies to improve performance, since $\frac{2}{3}$ is too much for a regret.

Note the independence of both Y_1 and Y_2 with respect to all other variables $\bar{\mathbf{P}}$. This means that even assuming a low level of dependency among labels, CLR may present a poor performance. In fact, there is only a single dependence, which is between Y_1 and Y_2.

In addition to being a high regret distribution for CLR, $\bar{\mathbf{P}}$ is the worst case distribution for the regret of the optimal solution for Hamming loss with respect to subset 0/1 loss:

$$\sup_{\mathbf{P}} r_s(\mathbf{h}^*_H) = R_s(\mathbf{h}^*_H) - R_s(\mathbf{h}^*_S) = \frac{1}{2}, \text{ when } \epsilon \to 0$$

where $\mathbf{h}^*_H$ is the optimal solution for Hamming loss on distribution $\bar{\mathbf{P}}$ and $\mathbf{h}^*_s$ for subset 0/1 loss (DEMBCZYŃSKI et al., 2012). Hence, $\bar{\mathbf{P}}$ simultaneously gives poor regret for Hamming loss optimizer and $\mathbf{h}^{\text{clr}}$ with respect to subset 0/1 loss.

4.2 A note about the average-case scenario

The studied worst-case scenarios (probability distributions) in which the RPC and CLR were analyzed, may seem rare in an average-case scenario practice at first glance, and not common in the average-case scenario however, defining precisely and objectively what is the "average-case scenario" is not a simple task. Firstly, the average case in the risk minimization of multi-label learning should define a probability distribution of label probability distributions. Defining a distribution of distributions is such a complex task that almost the whole Chapter 5 is concerned with succeeding in this task. Second, the average scenario is something that may vary from application to application. For instance, there exist researches that only concerned with multi-label problems of very low label density (extreme multi-label classification) and others that are only concerned with multi-label problems of high/low level of label dependence. If one tries to study multi-label learning in a one broad general scenario, where each specific multi-label scenario is taken with equal consideration without any a posteriori information, it should define a "Uniform distribution" that considers each case "evenly". Interestingly, the Uniform distribution that it is usually defined and used for this purpose does not take into account the "complexity/simplicity" associated with each possible sample/case/scenario. But why the simplicity of samples, objects, scenarios or hypothesis should be taken into account? The Occam's razor principle tells to choose the simplest hypothesis among all hypotheses consistent with the facts. Although not an irrefutable principle, it has been widely used in science and it is widely accepted that the Bayesian inference incorporates the Occam's razor into a more objective procedure for inference (JEFFERYS; BERGER, 1992; BLANCHARD; LOMBROZO; NICHOLS, 2018). The idea is that the simplest scenarios (probability distributions in our case), are, *a priori*, more probable. Since the all distributions defined in this chapter are simple (only a linear number of non-null values and these values can be easily computed), it may be that these distributions are more relevant than one might think.

There are theorems developed by Schöning e Pruim (2012) that support these claims, in which it is used a probability distribution called Universal probability distribution, a distribution that partly incorporates the idea of the Occam's razor and the Bayesian inference by quantifying the "randomness" (or the complexity) of an input object and using it to weight the probability of occurrence of these objects. In the end, the authors Schöning e Pruim (2012) have shown that the constructed Universal probability distribution gives an average time complexity equal to the worst time complexity. Hence, the worst case scenario is crucial when assuming a distribution such as the Universal probability distribution.

Chapter Conclusion

It has been revealed three factors highly impacting in a poor performance of RPC and CLR in the worst case scenario: The label cardinality and/or the mode of the distribution and the pairwise approach adopted by both. With this said, it is expected that the results presented in this chapter help researchers to be aware of the consequences of using the pairwise comparison approach done by RPC in multi-label problems of very low label cardinality. The results take one step closer to understanding the factors causing a good/bad performance on multi-label algorithms.

It is important to note that, despite all the "bad" results found towards RPC and CLR, RPC was proved to be risk minimizer for a particular loss function called Spearman rank correlation (Hüllermeier; Furnkranz, 2004). This loss function is for ranking problems, where the function receives the target ranking and the predicted rank as parameters, while Rank Loss receives the real labelling instead and the predicted ranking.

Further investigations can be done in order to improve the performance of CLR in the worst-case regret where possible improvements are better choosing the calibration threshold or changing the pairwise classifiers scope.

5 An experimental framework for evaluating loss minimization in multi-label classification via stochastic process

One major challenge Multi-Label Classification (MLC) faces, are the conditions for evaluating multi-label algorithms. Simplistic experimental setups based on artificial data may not capture crucial situations for analyzing these algorithms. This chapter introduces an experimental framework for evaluating multi-label algorithms by artificially generating the probabilistic label distributions. Although studies about the impacts of the dependence among labels based on experiments that artificially generate label distributions are also present in the literature (DEMBCZYŃSKI et al., 2012; WAEGEMAN et al., 2014), there are some important aspects that should be taken into account when generating artificial data for MLL that are not considered in these studies. For instance, their experimental setup does not allow to fully control the level of dependence among labels.

Following the Dembczyński et al. (2012) perspective and the idea probabilistic multi-label classifiers (see Section 2.1), it was taken into consideration that the prediction phase of some multi-label classifiers can be decomposed into two parts: the first part where an estimate of the probability distribution of labels is made based on the feature vector, and the second part where an actual labelling is predicted based only on the estimated probability distribution of labels of the first part (see Figure 1 of Section 2.1). The first part is responsible for finding out relations between the labels and the features of a given observation. Therefore, it was considered that multi-label methods can output an estimate of the probability distribution of labels, or part of it, alongside with a predicted labelling.

Inspired by Waegeman et al. (2014) and Jiang et al. (2014), the experimental framework proposed in this chapter deals with simulated probability distributions. Its objective is to assist studies about the relation between the average performance of multi-label methods and the level of dependence among labels. Indeed, the experimental framework can be used to simulate a dataset distribution with a specified expected level of dependence among labels, so researchers can produce multiple dataset distributions, each one with a different level of dependence, and compare the average performance of multiple multi-label methods. The framework provides a better control of the dependence among labels than other experimental setups in the literature by making fewer assumptions in the simulation. It also takes into account another important aspect, the difficulty of the problem, which is related to how difficulty is to make a prediction and has a close relationship with entropy. The difficulty of the problem represents the amount of valuable information available for predicting or the randomness of the labels. Some studies try to incorporate this aspect by applying noise to the dataset features. The dependency among labels is widely analyzed in literature (WAEGEMAN et al., 2014; DEMBCZYŃSKI; CHENG; HÜLLERMEIER, 2010; DEMBCZYŃSKI et al., 2012), while the difficulty of the problem is much less investigated (SENGE; COZ; HÜLLERMEIER, 2013; TOMÁS et al., 2014).

Therefore, the contribution in chapter is proposing an experimental framework for evaluating MLC methods based on generating artificial label distributions which allows to:

- Generate a wider variety of probability label distributions than previous works by making fewer assumptions with respect to the simulated labels.
- Control the level of dependence among labels.
- Control the difficulty of the problem.

The framework incorporates these three aspects simultaneously, allowing studying multi-label algorithms inside the probabilistic framework of Dembczyński et al. (2012). As far as it is concerned, no other work has addressed them simultaneously yet.

Using this experimental framework, an experimental case study was conducted and some interesting relations among the tested methods were revealed, which are listed:

- **F-measure and Jaccard distance optimizers:** An algorithm designed to maximize F-measure is a good approximation to optimize Jaccard distance;
- **Calibrated label ranking by pairwise comparison and binary relevance:** Both methods have equal performance when the difficulty level is high or the level of label dependence is very low.
- **Classifier Chains and Probabilistic Classifier Chains:** Both have a similar performance in all tested metrics. In addition, they are the best in subset 0/1 loss.

The remainder of this chapter is organized as follows. Section 5.1 reviews and compares related experimental setups in works that used synthetic data for evaluating multi-label methods. In Section 5.2 is described the proposed experimental framework itself and discussed the use of the beta distribution to simulate data for loss minimization problems. Section 5.3 discusses results of an experimental case study conducted. The chapter ends with our findings and final observations about this framework.

5.1 Multi-label experimental setups

The performance of MLC methods is occasionally evaluated by experimental studies on synthetic datasets. Evaluation on synthetic datasets is usually done by generating instances whose features are random variables and labels are non-deterministic functions of the features (DEMBCZYŃSKI et al., 2012; ZHANG; PEÑA; ROBLES, 2009; TOMÁS et al., 2014). A simple example consists of generating each feature of an instance $(\mathbf{x},\mathbf{y})$ according to the normal distribution and randomly associating labels with a probability defined by the logistic function:

$$\mathbf{P}(y_i = 1|\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}\cdot\mathbf{x}}},$$

where $\mathbf{w}$ is a vector of real constant values with the same size of $\mathbf{x}$ and $\mathbf{w} \cdot \mathbf{x}$ represents the inner product between $\mathbf{w}$ and $\mathbf{x}$. In practice, even knowing that the whole probability distribution of n labels takes $2^n - 1$ independent values (each combination of labels can independently have its own probability value), experiments are performed using a linear number (much smaller than $2^n - 1$) to generate synthetic labels (TOMÁS et al., 2014; ZHANG; PEÑA; ROBLES, 2009; DEMBCZYŃSKI et al., 2012). For example, Zhang, Peña e Robles (2009) and Dembczyński et al. (2012) apply their methodology on artificially generated datasets, where the vector of features is uniformly random distributed in a hypersphere and uses one rule for each label to generate all $2^n - 1$ combinations of labels. When using only these specific n rules, the simulation is limited to a linear number of variables.

Artificial multi-label datasets are generated in the work of Noh, Song e Park (2004) with the purpose of testing feature selection embedded in multi-label methods. First, the combinations of labels are evenly generated and then features are created as a function of the given combination of labels. For example, in one of the datasets a real-valued feature x follows a normal distribution. Assuming labels y_1 and y_2 are given, the parameters of the distribution vary according to the following rule:

$$x \sim \begin{cases} N(0,1) & \text{if } y_1 = y_2 = 1, \\ N(0.5,1) & \text{if } y_1 = y_2 = 0, \\ N(1,1) & \text{if } y_1 \neq y_2. \end{cases}$$

Using the Bayes' formulae and knowing that combinations of labels are distributed evenly, it can be shown that this particular experiment suffers from correlated variables. The probability $\mathbf{P}_{\mathbf{x}}(y_1 = 1, y_2 = 1) = \frac{\sqrt{e}}{\sqrt{e}+e^{3/8+x/2}+2e^{x}}$ while $\mathbf{P}_{\mathbf{x}}(y_1 = 0, y_2 = 1) = \frac{e^{x}}{\sqrt{e}+e^{3/8+x/2}+2e^{x}}$, which are clearly dependent on each other, limiting the number of possible probability distributions. A simulation of 1 million values for feature x reveals that these two probability values have a Pearson correlation of approximately -0.98. Although this experiment can be extended to more labels and the means of the used normal distributions varied, it is hard to see what happens with the properties of the label distribution, such as the level of label dependence. This is a limitation of the experiment that is not imposed in the work proposed framework, or at least not as strong as imposed in the work by Noh, Song e Park (2004).

In the work of Ye et al. (2012) an artificial dataset is generated by first assuming that the conditional distribution of the feature vector given a single label follows a mixture of Gaussian distributions and then randomly generating instances according to this distribution. They test a set of algorithms in multiple datasets varying the number of instances, number of features and the average distance between positive and negative instances. They focus on binary classifications, consequently label dependencies are not considered explicitly, although multi-label datasets with a possible correlation among labels can be generated from the same artificial data. The possibility of changing the average distance between positive and negative instances is an interesting aspect of their approach because it allows to somehow control the difficulty of the classification problem.

The authors Pereira et al. (2018) analyze 16 distinct multi-label evaluation measures in order to aid researchers when choosing a subset of evaluation measures. The authors consider multi-label evaluation measures as random variables and then estimate the Pearson Correlation between each pair of multi-label measures. For this purpose, they performed experiments of 16 multi-label methods on 11 multi-label datasets. The analysis showed which measures have strong correlation with each other, e.g Example-Based Accuracy has a strong correlation with Example-Based F-measure. Interestingly, the analysis presented in this chapter is very consistent with the analysis of Pereira et al. (2018), although the purpose here is not only to analyze measures, but also, and mainly, multi-label methods.

The authors of Dembczyński et al. (2012) undertook an experimental study to analyze the behavior of methods when label dependence changes. Three types of experiments are performed, each one exploring a different type of label dependence, the marginal independence, the conditional independence and the conditional dependence, as named by the authors. Even though the proposed experimental framework focuses only on the conditional dependencies among labels, it allows a much deeper and generic experimental analysis of this type of dependence than the work of Dembczyński et al. (2012).

A simulation on the scores given by base binary classifiers used by a multi-label method is conducted by Jiang et al. (2014). They assume that scores of binary classifiers are distributed according to a Beta distribution and directly generate data from the marginal distributions of the labels not focusing on the label correlation, which is a strong limitation when studying multi-label algorithms in general. The beta distribution was chosen because it is bounded by 0 and 1, the mean can be defined at any value in $(0, 1)$ and the variance can be fully controlled as desired (0 to 1/4). It also includes common distributions as special cases, like the uniform distribution and the Bernoulli distribution.

Contrasting to the work of Jiang et al. (2014), the framework proposed in this chapter addresses the main limitation of their study by considering all nodes of probability trees to take into account label dependencies to better evaluate multi-label methods in loss minimization.

Waegeman et al. (2014) conducted two classes of experiments to empirically compare four multi-label methods. The first class assumed independence of labels, while the second class assumed a model with strong label dependencies. Instead of artificially generating observations with features and then assigning labels based on the features, the authors directly generated labels from a predefined distribution of labels $\mathbf{P}(\mathbf{y})$. In the second class, the authors assume a model that the following restriction holds

$$\mathbf{P}(y_i|y_1, ..., y_{i-1}) = \frac{1}{1 + \exp(-\sum_{j=1}^{i-1} 2w_{ij}(y_j - \frac{1}{2}) - w_{i0})}, \tag{5.1}$$

where all $w_{ij} \sim N(1, 3)$ and $w_{i0} \sim N(1, 3)$. The labellings are generated by the chain rule of probability

$$\mathbf{P}(\mathbf{y}) = \prod_{i=1}^{m} \mathbf{P}(y_i|y_1, ..., y_{i-1}).$$

In the experiments the number of labels was set to 25 and the number of observations varied from 5, 10, 20, 30, 40, 50, 75, 100, 200, 500, 1000, 2000, 5000, 10000. For each one, the experiments were repeated for 30 different values of w_i. Their experimental setup differs from the one proposed in this chapter mainly in the level of label dependence which is better controlled in this framework. Using a metric for calculating the level of label dependence, defined later in Equation (5.3) at the next section, a 12 label problem following the distribution presented in Equation (5.1) shows an average value of 0.398. Using the framework defined in this chapter, it is possible to choose a level of label dependence up to 0.5.

5.2 The Experimental Framework

The experimental framework simulates several $\mathbf{P}(\mathbf{Y}|\mathbf{X} = \mathbf{x})$ distributions and gives them to the multi-label classifiers. Consequently, no model needs to be induced from the training dataset, and no estimate of $\mathbf{P}$ needs to be made, making the feature vector unnecessary in the prediction process. Hence, the framework does not evaluate the training process of a multi-label method, but only the final part of the prediction process, where a multi-label classifier predicts a labelling based only on the given distribution of labels.

A probability tree diagram can be used to explain the framework. A probability tree is a weighted binary tree representing a sequence of conditional events, in this case, a labelling. Each level of the tree represents a label and each node represents the occurrence (or absence) of the respective label given the previous nodes (labels). An example considering only two labels (C and D) is shown in Figure 2.

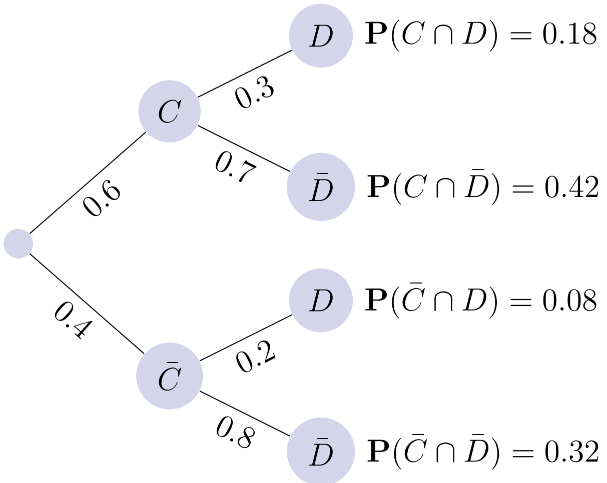

D
$\mathbf{P}(C \cap D) = 0.18$
C
0.3
0.7
$\bar{D}$
$\mathbf{P}(C \cap \bar{D}) = 0.42$
0.6
0.4
D
$\mathbf{P}(\bar{C} \cap D) = 0.08$
$\bar{C}$
0.2
0.8
$\bar{D}$
$\mathbf{P}(\bar{C} \cap \bar{D}) = 0.32$

Figure 2 – An example of a probability tree with events C and D

Note that a path from the root to a leaf node represents the joint event with probability equals to the product of the path weights. For example, the probability of occurrence of event C is written as $\mathbf{P}(C) = 0.6$. The probability that D occurs given C is written as $\mathbf{P}(D|C) = 0.3$ and, therefore, the probability of such instance being associated with labels C and D is 0.18.

For each level $i \in (1, n)$ of the tree there are 2^i edges corresponding to the probabilities of occurrence (or absence) of label i given all possible combinations of the previous labels. Note, however, that half of the edges are determined by complementary probability, since the probability of absence can be given as a function of the probability of occurrence. Therefore, from now on, only the probabilities with respect of the occurrence of events are considered and discussed. For each $j \in (1, r)$, where $r = 2^{i-1}$, let B_{ij} be the value of the j-th node of the i-th level and let each level represent a label. In the example of Figure 2, this means that $B_{11} = \mathbf{P}(\mathbf{Y}_i = 1) = 0.6$ and $B_{22} = \mathbf{P}(\mathbf{Y}_2 = 1|\mathbf{Y}_1 = 0) = 0.2$. Given two vectors of constants $\boldsymbol{\mu} = (\mu_1, ..., \mu_n) \in [0,1]^n$ and $\boldsymbol{\theta} = (\theta_1, ..., \theta_n) \in [0,1]^n$, it is assumed that each variable B_{ij} is a random variable that is independently distributed from a probability density function of mean μ_i and variance $\mu_i(1 - \mu_i)\theta_i$. Each μ_i from vector $\boldsymbol{\mu}$ is a random variable that is i.i.d from a distribution of an arbitrary mean $\hat{\mu}$ and variance $\hat{\mu}(1 - \hat{\mu})\hat{\boldsymbol{\theta}}$. Analogously, each θ_i is a random variable that is i.i.d from a distribution of an arbitrary mean $\bar{\mu}$ and variance $\bar{\mu}(1 - \bar{\mu})\bar{\boldsymbol{\theta}}$.

Note that not all labels follow the same distribution. For example, $B_{21}B_{11} + B_{22}(1 - B_{11})$, which corresponds to the marginal distribution of label 2, in general, is not the same distribution as B_{11}, which corresponds to the marginal distribution of label 1. However, some properties remain the same for all labels, for instance, the expected density of each label:

$$\mathbf{P}_{\mathbf{x}}(y_i = 1) = \hat{\mu}, \text{ for all } 1 \leq i \leq n \quad \text{(proof in Appendix C.1)}. \tag{5.2}$$

In order to evaluate a multi-label method in this experimental framework, the method must be expressed as a function of the distribution of labels. Examples are found in Section 2.1 where the prediction of multi-label classifiers are expressed as functions of the distribution of the labels. Although only transformation based methods are presented, a few specific algorithm adaptation methods can also be expressed as a function of the distribution of labels, for instance, according to Cheng e Hüllermeier (2009), Younes et al. (2011), Zhang e Zhou (2013), ML-KNN (Min-Ling Zhang; Zhi-Hua Zhou, 2005) has the same predictions as BR whose binary classifiers are a modified KNN algorithm. In the words of Cheng e Hüllermeier (2009), "ML-KNN is a binary relevance learner, i.e., it learns a single classifier for each label. However, instead of using the standard k-nearest neighbor classifier as a base learner, it implements the single classifier by means of a combination of KNN and Bayesian inference.". Also, as claimed by Decubber et al. (2019), a multi-layer neural network whose output layer contains one neuron for each label and each output neuron uses a logistic activation function, is the equivalent of the BR method. In this case, the ith output neuron is an estimate of $\mathbf{P}(\mathbf{Y}_i|\mathbf{x})$ and hidden layers act as feature extractors of $\mathbf{x}$. Note that the predictions of these two methods are only expressed as a function of the distribution of labels because they can be actually viewed as a transformation method. It is possible that any multi-label adaptation method can be somehow viewed as a transformation based method. This is not commonly believed in the literature, but it is hard to prove whether it is true or false.

The experimental procedure of the framework is presented in Algorithm 3 with four parameters, ϕ, γ, $\hat{\mu}$, and $\bar{\theta}$:

- The parameter γ is used as a measure of dependence between labels;
- ϕ is used as a measure of difficulty of the problem;
- $\hat{\mu}$ is the average label cardinality. A high value means that simulations will in average produce probability distributions where is expected that labellings have a number of positive labels (i.e $\sum_i \mathbf{P}(Y_i = 1)$ is high);
- $\bar{\theta}$ is a parameter related to the variability among the variance of tree levels. A low value means that nodes of the same tree level vary equally (same variance).

```
input  : A multi-label method M;
         Number of labels $n$;
         Number of iterations $k$;
         Loss function L;
         Real values $\hat{\mu}$ and $\bar{\theta}$ such that $0 < \hat{\mu} < 1,\ 0 < \bar{\theta} < 1$;
         Real values $\phi$ and $\gamma$ such that $\phi^2 + \gamma^2 < \hat{\mu}(1-\hat{\mu})$.
output : Estimated Expected loss of M for the given loss function.

Consider a probability tree $\mathbf{T}$ of $n$ events in which $B_{ij} \in \mathbf{T}$, for any $1 \leq i \leq n$ and
  $1 \leq j \leq 2^{i-1}$, represents the conditional probability of occurrence of label $i$ given
  a particular combination $j$ of labels in $\{1, 2, ..., i-1\}$;
$r \leftarrow 0$;
$\bar{\mu} \leftarrow \frac{\gamma^2}{\gamma^2+\phi^2}$;
$\hat{\theta} \leftarrow 1 - \frac{\gamma^2+\phi^2}{\hat{\mu}(1-\hat{\mu})}$;
repeat $k$ times
    $\mu_1, ..., \mu_n \leftarrow$ Random($\hat{\mu}$,$\hat{\theta}$);
    $\theta_1, ..., \theta_n \leftarrow$ Random($\bar{\mu}$,$\bar{\theta}$);
    foreach $B_{ij} \in \mathbf{T}$ do $B_{ij} \leftarrow$ Random($\mu_i$,$\theta_i$);
    $\hat{\mathbf{y}} \leftarrow$ M($\mathbf{T}$);
    foreach $\mathbf{y} \in \mathcal{Y}$ do
        $r \leftarrow r +$ L($\mathbf{y}$,$\hat{\mathbf{y}}$) $\cdot$ JointProb($\mathbf{y}$,$\mathbf{T}$);
    end
end
return $r/k$
```

Algorithm 3 : Process for evaluating a multi-label method by simulating the distribution of labels. Function `Random(`μ,θ`)` returns a random value in $[0, 1]$ from an arbitrary distribution of mean μ and variance $\mu(1-\mu)\theta$. Function `JointProb(`$\mathbf{y}$,$\mathbf{T}$`)` returns the joint probability of labelling $\mathbf{y} \in \mathcal{Y}$ in the distribution of labels $\mathbf{T}$.

An example of a random density distribution for each node B_{ij} (function `Random(`$\cdot$,$\cdot$`)` in Algorithm 3) can be the Beta distribution. Three examples of Beta distributions are given in Figure 3.

The value of γ^2 is defined as the mean of the expected value of the average quadratic difference between each pair of nodes of a specific level:

$$\begin{aligned}\gamma^2 =& \mathbb{E}\left[\frac{1}{n-1}\sum_{i=2}^{n}\frac{1}{\binom{2^{i-1}}{2}}\sum_{(j,k):j\neq k}(B_{ij}-B_{ik})^2\right]\\ =& \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\theta}}\left[\frac{1}{n-1}\sum_{i=2}^{n}\frac{1}{\binom{2^{i-1}}{2}}\sum_{(j,k):j\neq k}\mathbb{E}\left[(B_{ij}-B_{ik})^2|\boldsymbol{\mu},\boldsymbol{\theta}\right].\right]\end{aligned} \tag{5.3}$$

To calculate the right-hand side of the equation, it is necessary to calculate $\mathbb{E}\left[(B_{ij}-B_{ik})^2|\mu_i,\theta_i\right]$. The formula for calculating the variance of an arbitrary random variable X gives us a way to calculate it:

$$\mathrm{Var}\left[X\right] = \mathbb{E}\left[X^2\right] - \mathbb{E}\left[X\right]^2. \tag{5.4}$$

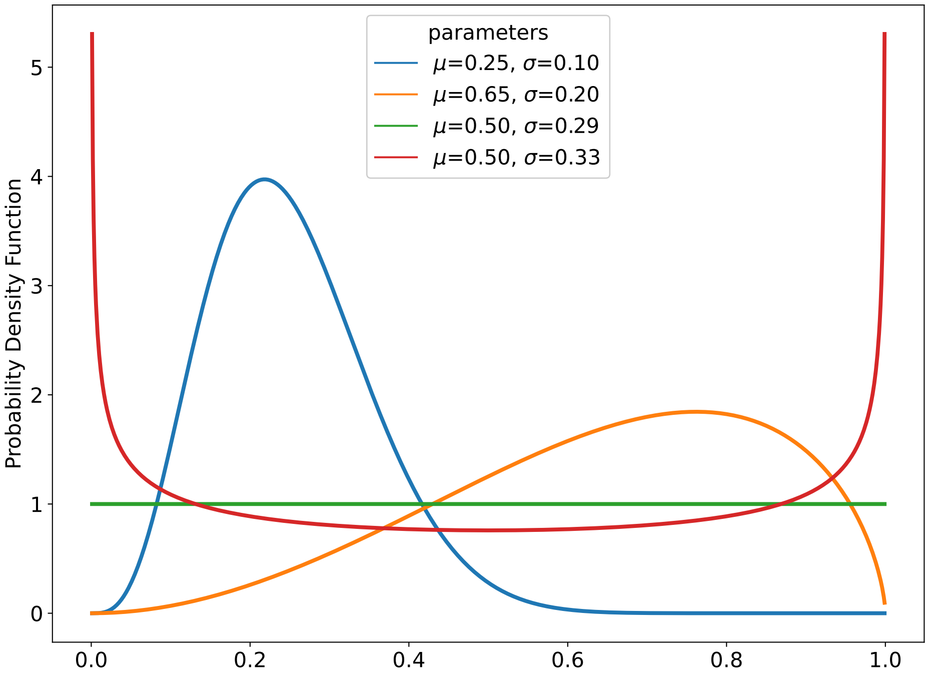


Figure 3 – Four examples of different parameters (mean and standard deviation) for the Beta distribution. Note that for a specific pair of parameters, the Beta distribution becomes the uniform distribution.

From (5.4) we have that

$$\mathbb{E}\left[X^2\right] = \mathrm{Var}\left[X\right] + \mathbb{E}\left[X\right]^2.$$

If X is substituted by $B_{ij} - B_{ik}$ when conditioned in μ_i and θ_i, then $\mathbb{E}\left[X^2\right] = \mathbb{E}\left[(B_{ij} - B_{ik})^2|\mu_i, \theta_i\right]$ is given by:

$$\begin{aligned} &\mathrm{Var}\left[B_{ij} - B_{ik}|\mu_i, \theta_i\right] + \mathbb{E}\left[B_{ij} - B_{ik}|\mu_i, \theta_i\right]^2 \\ =& \mathrm{Var}\left[B_{ij} - B_{ik}|\mu_i, \theta_i\right] + (\mu_i - \mu_i)^2 \\ =& \mathrm{Var}\left[B_{ij} - B_{ik}|\mu_i, \theta_i\right]. \end{aligned}$$

Since $B_{ij} - B_{ik}$ are independent given μ_i and θ_i, therefore

$$\begin{aligned} \mathbb{E}\left[(B_{ij} - B_{ik})^2|\mu_i, \theta_i\right] =& \mathrm{Var}\left[B_{ij}|\mu_i, \theta_i\right] + \mathrm{Var}\left[B_{ik}|\mu_i, \theta_i\right] \\ =& \mu_i(1 - \mu_i)\theta_i + \mu_i(1 - \mu_i)\theta_i \\ =& 2\mu_i(1 - \mu_i)\theta_i. \end{aligned} \tag{5.5}$$

Substituting (5.5) in (5.3), it can be shown that

$$\begin{aligned}
\gamma^2 =&\mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\theta}}\left[\frac{1}{n-1}\sum_{i=2}^{n}\frac{1}{\binom{2^{i-1}}{2}}\sum_{(j,k):j\neq k}2\mu_i(1-\mu_i)\theta_i\right]\\
=&\mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\theta}}\left[\frac{1}{n-1}\sum_{i=2}^{n}\frac{2\mu_i(1-\mu_i)\theta_i}{\binom{2^{i-1}}{2}}\binom{2^{i-1}}{2}\right]\\
=&\frac{2}{n-1}\sum_{i=2}^{n}\mathbb{E}_{\mu_i,\theta_i}\left[\mu_i(1-\mu_i)\theta_i\right]\\
=&\frac{2}{n-1}\sum_{i=2}^{n}\hat{\mu}(1-\hat{\mu})(1-\hat{\theta})\bar{\mu}\\
=&2\hat{\mu}(1-\hat{\mu})(1-\hat{\theta})\bar{\mu}
\end{aligned} \tag{5.6}$$

The difficulty of the problem ϕ is related to the randomness of the labels, and have a close relation to the entropy of a distribution. For instance, if the probability of presence of each label falls very close to 0 or 1, then the problem is said to have low difficulty, because its has a low risk when saying that a label is (not) present. Let $\bar{B}_i = \frac{1}{2^i-1}\sum_{j=1}^{2^i-1}B_{ij}(1-B_{ij})$ for any i. The value of ϕ^2 is defined as the expected value of the geometric mean over $\bar{B}_1, ..., \bar{B}_n$:

$$\begin{aligned}
(\phi^2)^n &= \mathbb{E}\left[\prod_{i=1}^{n}\bar{B}_i\right]\\
&= \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\theta}}\left[\prod_{i=1}^{n}\mathbb{E}[\bar{B}_i|\mu_i,\theta_i]\right]
\end{aligned} \tag{5.7}$$

The expected value of $B_{ij}(1-B_{ij})$ for any i, j when given μ_i and θ_i is

$$\begin{aligned}
\mathbb{E}\left[B_{ij}(1-B_{ij})|\mu_i,\theta_i\right] &= \mu_i - {\mu_i}^2 - \mathrm{Var}\left[B_{ij}|\mu_i,\theta_i\right]\\
&= \mu_i(1-\mu_i) - \mu_i(1-\mu_i)\theta_i\\
&= \mu_i(1-\mu_i)(1-\theta_i)
\end{aligned} \tag{5.8}$$

From equations (5.7) and (5.8), one may have that

$$\begin{aligned}
(\phi^2)^n &= \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\theta}}\left[\prod_{i=1}^{n}\mu_i(1-\mu_i)(1-\theta_i)\right]\\
&= \prod_{i=1}^{n}\hat{\mu}(1-\hat{\mu})(1-\hat{\theta})(1-\bar{\mu}),
\end{aligned}$$

from which it is concluded that:

$$\phi^2 = \hat{\mu}(1-\hat{\mu})(1-\hat{\theta})(1-\bar{\mu}). \tag{5.9}$$

One may express $\bar{\mu}$ and $\hat{\theta}$ in terms of ϕ and γ by solving equations (5.6) and (5.9):

$$\bar{\mu} = \frac{\gamma^2}{\gamma^2+\phi^2} \tag{5.10a}$$

$$\hat{\theta} = 1 - \frac{\gamma^2 + \phi^2}{\hat{\mu}(1 - \hat{\mu})} \tag{5.10b}$$

In algorithm 3, one can use the beta distribution as a base for generating random values. The usual parameters of the beta distribution, say α and β, can be written in terms of the mean and variance, say μ and σ^2, as following (SULAIMAN et al., 1999):

$$\begin{gathered}\alpha = \left(\frac{\mu(1-\mu)}{\sigma^2} - 1\right)\mu \\ \beta = \left(\frac{\mu(1-\mu)}{\sigma^2} - 1\right)(1-\mu) \\ \text{for } 0 < \mu < 1 \text{ and } \sigma^2 < \mu(1-\mu)\end{gathered}$$

Through such parametrization one can analyze the effects of the dependence among labels and the difficulty of the problem. The major benefits of the described evaluation method are listed below:

- Lesser assumptions with respect to label distributions than previous works (see Section 5.1), in such a way that nodes are generated independently of each other. By using an independent random variable for each probability node of the tree, it is guaranteed that any possible distribution of labels can be generated and nodes are independent of each other.
- Two important factors of MLC problems are controlled. In this way, it is easy to verify if a multi-label method is sensible to dependence among labels and makes it possible to analyze the behavior of multi-label methods by simulating different configurations of datasets.

Given a probability tree $\mathbf{T}$ of n labels, the computational cost of a single simulation in Algorithm 3 is $\mathcal{T}(n) + M(\mathbf{T}) + \mathcal{R}(\mathbf{T})$, where $\mathcal{T}(n)$ is the cost of generating a probability tree of n labels, $M(\mathbf{T})$ is the cost of multi-label method M making a prediction, and $\mathcal{R}(\mathbf{T})$ is the cost of calculating the risk for a single prediction. In general, it is possible to say that:

- $\mathcal{T}(n) = \Omega(2^n)$, since 2^n real values need to be generated from a random distribution;
- $\mathcal{R}(\mathbf{T}) = \Omega(2^n)$, since it is a sum of 2^n values;
- If one uses the brute force algorithm (testing all 2^n labellings), $M(\mathbf{T}) = 2^n\mathcal{R}(\mathbf{T}) = \Omega(2^{2n})$.

Hence, Algorithm 3 has a very high computational cost of at least $\Omega(2^{2n})$ when the brute force algorithm is used, making it unfeasible for high values of n. This can be highly reduced at specific combinations of multi-label classifier and multi-label metric. Consider the evaluation of CC in terms of subset 0/1 loss. Since subset 0/1 loss is only zero when both predicted and target labellings are equal, and 1 otherwise, the risk of an arbitrary labelling $\hat{y}$ with respect to subset 0/1 loss can be easily calculated by $1 - \mathbf{P}(\mathbf{Y} = \hat{y}|\mathbf{x})$, which is much faster computed than a sum of 2^n values. To compute the CC prediction, only n values of the probability tree needs to be generated: $\{\mathbf{P}(Y_1 = 1),\ \mathbf{P}(Y_2 = 1|Y_1 = y_1^{\text{CC}}),\ \ldots,\ \mathbf{P}(Y_n = 1|Y_1 = y_1^{\text{CC}}, \ldots, Y_{n-1} = y_{n-1}^{\text{CC}})\}$, removing the need of generating all 2^n values of the probability tree.

5.3 An Application of the Experimental Framework

To demonstrate the effectiveness of the experimental framework, a case study is carried out. To this end, the expected losses, considering the four loss functions presented in Section 2.2, were estimated for five multi-label algorithms: BR, DBR, CLR, CC and PCC. [1]

For comparison reasons, an optimal classifier for Jaccard distance is obtained by exhaustive search. Note that it is only possible because the number of labels is not very high, otherwise, it would be impracticable due to the exponential growth of possible labellings. In order to determine the values of $\bar{\mu}$ and $\hat{\theta}$ by equations (5.10a) and (5.10b), an arbitrary value from 0 to 1 must be chosen for $\hat{\mu}$. In these experiments, $\hat{\mu}$ is fixed to $\frac{1}{2}$. The methods were tested with 9 distinct values for $\phi \in \{0.05, 0.1, ..., 0.45\}$. From equation (5.10b), the inequality $\gamma^2 < \frac{1}{4} - \phi^2$ must be true, therefore the tested values for γ were defined as a function of ϕ:

$$\gamma = \lambda\sqrt{\frac{1}{4} - \phi^2},\ \text{for } \lambda \in \{0.1, 0.2, ..., 0.9\} \cup \{0.025, 0.05, 0.99\}.$$

This gives a total of 108 distinct configurations for (γ, ϕ). For each combination of the parameter pairs, method and loss function, the expected losses were estimated over 10000 simulations based on probability trees composed of 12 labels. The case study is restricted to 12 labels only due to the high computational cost of computing the optimal solution for the Jaccard distance, which is usually not feasible in practice. Therefore, for a large number of labels, the exhaustive methods should be avoided and the experiments conducted exclusively on the feasible methods.

Instead of presenting the expected losses on their absolute value, they are normalized by dividing them by the Bayes error (the best solution value). Thus, the normalized values can be interpreted as how far the classification is from the global minimum loss. The multi-label method that optimizes the F-measure or the Jaccard distance are named Bayes F-measure and Bayes Jaccard distance, respectively.

[1] The source code in Python programming language is available at <https://github.com/Lucashsmello/mll-framework>

First, a summary of the results is presented on Table 1 and then some details are discussed. The summary is an average ranking of all classifiers for each metric. The ranking is done for each one of the 108 configurations, that is, for each tested (γ, ϕ) the rank 1 was given to classifier with lowest estimated risk and 7 for the highest. When a tie occur, the average rank is given to all classifiers involved in the tie.

To optimize Hamming loss it is enough to only consider the marginal distributions as shown in Dembczyński, Cheng e Hüllermeier (2010), which is done by BR.

Table 1 – Average rank of all tested Multi-label methods over 108 experiments.

	Hamming loss	Subset 0/1 loss	F-measure	J. distance
BR	1.00	4.16	3.12	3.06
DBR	6.55	3.82	6.99	6.99
CC	4.72	2.07	5.50	5.50
PCC	4.54	1.00	5.40	5.12
Bayes JD	3.90	4.40	2.00	1.00
Bayes FM	4.95	6.48	1.00	2.00
CLR	2.31	6.04	3.87	4.31

Clearly the results in Table 1 show that CC is not designed to optimize Hamming loss, Jaccard distance and neither F-measure, opposed to some works that suggests that CC optimizes Hamming loss (DOPPA et al., 2014; ZHANG; ZHOU, 2013). This may happen because CC achieves a good Hamming loss in some special cases, but in average, it does not. Note that the F-measure optimizer achieves a great performance on Jaccard distance. While no efficient optimizer is known for Jaccard distance, the F-measure optimizer, which runs in polynomial time (DEMBCZYŃSKI et al., 2013), can be used as an approximation.

Some results are presented in the form of pairs of graphs in figures from 4 to 7. Each pair represents the extremes with respect to difficulty: the left shows the relative expected loss for multiple values of γ and $\phi = 0.05$, while the right shows the same but with $\phi = 0.45$. The extreme values for ϕ were chosen to be shown and discussed here, because they can be used to represent all other experiments, since the intermediate values present results that are almost an weighted average of the extreme values. Nevertheless, the results for all values of ϕ are included in Appendix C.2, C.3, C.4 and C.5.

From the results, two interesting observations are made regarding subset 0/1 loss:

- CC is a good approximation for optimizing subset 0/1 loss. It needs only n parameters of the conditional joint distribution over labels and in all experiments it was the best one behind PCC, which needs 2^n parameters for the same distribution. Clearly this happens due to the fact that CC is a greedy heuristic for finding the labelling with the highest probability, as discussed by Dembczyński, Cheng e Hüllermeier (2010).

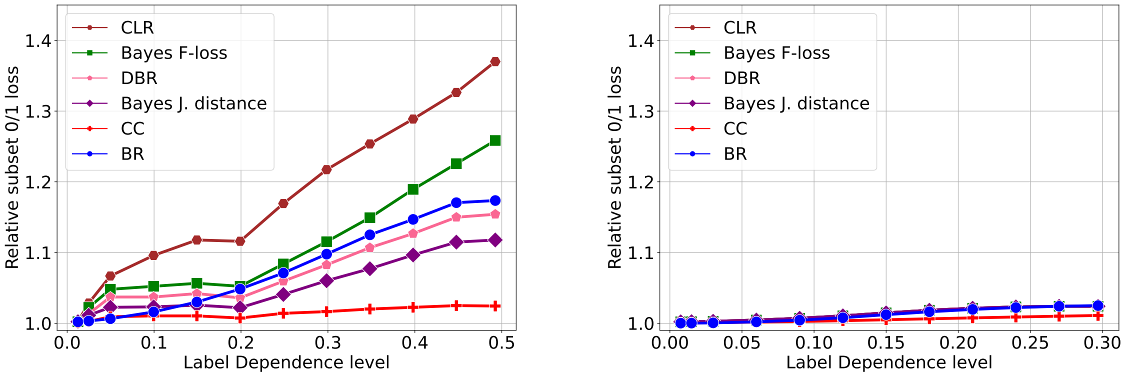


Figure 4 – Graphs of estimated expected subset 0/1 loss for all methods divided by the best one (PCC) when $\phi = 0.05$ (left) and $\phi = 0.45$ (right). The value of γ is presented on the horizontal axis. In the right figure, all classifiers, except for CC, resulted in almost the same performance, hence all of them collapsed to the same line.

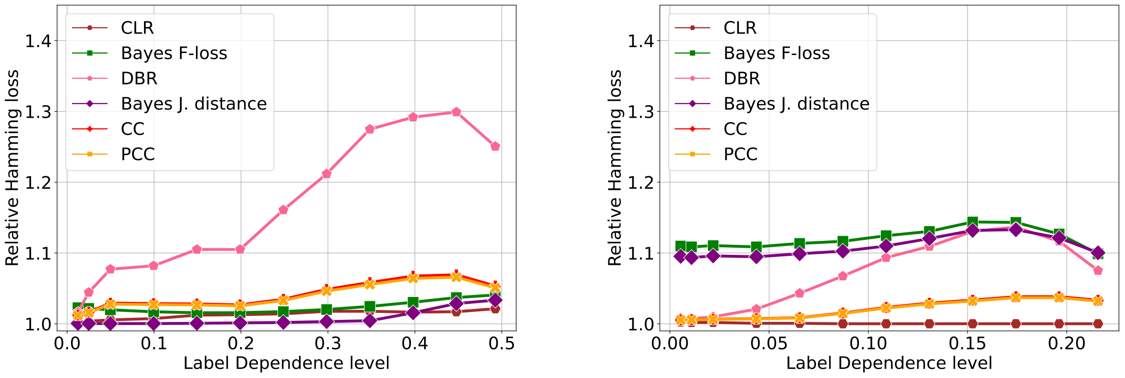


Figure 5 – Graphs of estimated expected Hamming loss for all methods divided by the best one (BR) when $\phi = 0.05$ (left) and $\phi = 0.45$ (right). The value of γ is presented on the horizontal axis. The results from BR method corresponds to value 1.

- The optimal algorithm for F-measure is the worst among the tested ones. Even considering the correlation between labels, this algorithm does not achieve a better expected loss than BR. Indeed, Subset-loss, a very strict measure, seems more related to Jaccard distance than to F-measure loss by the fact that Jaccard distance is stricter than F-measure loss, i.e, always gives a higher or equal loss for the same prediction except when the prediction is fully correct or fully incorrect (see equation 2.10).

Note that exploiting dependence among labels does not imply better performance (e.g, CC has a higher expected loss in F-measure than BR, even when the level of label dependence is high). In all results, note that CC and PCC had a very similar performance, reinforcing

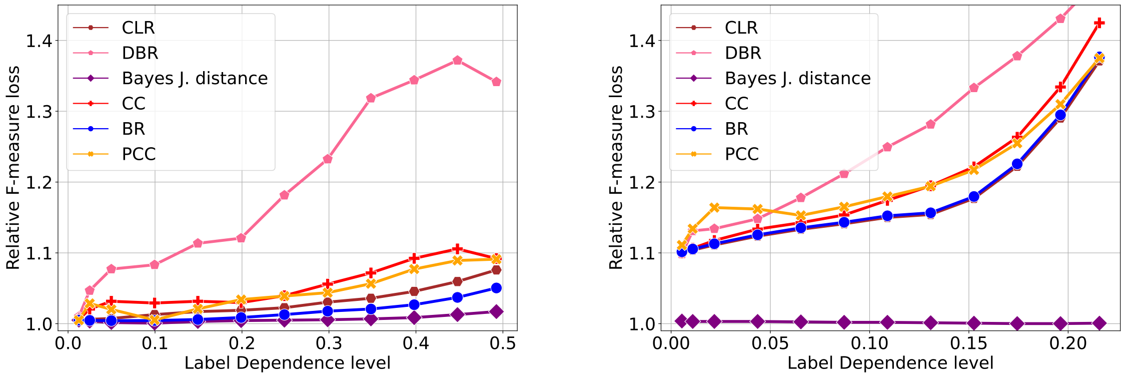


Figure 6 – Graphs of estimated expected F-measure loss for all methods divided by the best one (Bayes F-measure) when $\phi = 0.05$ (left) and $\phi = 0.45$ (right). The value of γ is presented on the horizontal axis. The results from F-measure optimizer method corresponds to value 1.

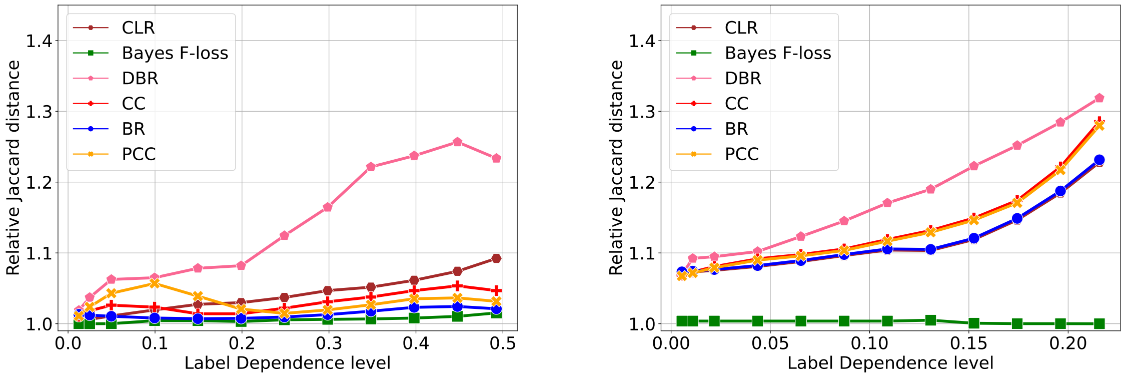


Figure 7 – Graphs of estimated expected Jaccard distance for all methods divided by the best one (Bayes Jaccard distance) when $\phi = 0.05$ (left) and $\phi = 0.45$ (right). The value of γ is presented on the horizontal axis. The results from Jaccard distance optimizer method corresponds to value 1.

the idea that CC is an approximation algorithm of PCC. The expected F-measure loss was proved that it can be optimized efficiently using only $\mathcal{O}(n^2)$ parameters of the joint label distribution (DEMBCZYŃSKI et al., 2013) whereas no efficient algorithm for optimizing Jaccard distance is known.

The results in Figures 6 and 7 show a very strong relation between F-measure and Jaccard distance optimizers. Note that the estimated values of the expected losses for both optimizers are almost indistinguishable on the graphs of Figures 6 and 7. The results suggest that F-measure optimizer, which has a known efficient algorithm, can be used as an approximation algorithm for optimizing Jaccard distance, for which no efficient optimizer is known. Pereira et al. (2018) conclude that F-measure and Jaccard distance (a.k.a, the complement of Example based accuracy) have a strong correlation. The results in this chapter show that along with the strong correlation, the optimal solutions for both measures are similar.

Note that CLR had always a higher expected loss than BR. This occurs in all experiments conducted in this section, which one can infer that CLR is not a good choice for any measure tested, but may be for other measures not used here.

From the graphs of subset 0/1 loss, F-measure and Jaccard distance, there are clear evidences that DBR exploits a type of dependency among labels that is not beneficial for optimizing none of these metrics: The performance greatly becomes relatively worse when label dependence level is increased. Moreover, DBR is the method with the most cases at the highest average loss. There is no evidence from the results to suggest the usage of DBR in any case, if one is concerned with optimizing one of the four multi-label metrics considered. Again, it is possible that DBR optimizes a multi-label metric that was not considered in experimental study. Furthermore, it may be possible that the major strength of DBR relies on being tolerant to estimation errors. As its creators (MONTAÑES et al., 2014) suggested, the redundancy introduced by DBR may improve performance in practice, as shown in many branches of machine learning, such as in ensemble of classifiers. As probability distributions of labels are assumed to be perfectly known in the framework, the supposed tolerance against errors in estimating the distribution is not taken into account.

The level of label dependence is not the only relevant variable for MLC. In figures 5, 6 and 7 one may see that a high level of difficulty makes some methods further from the optimal, even when there is no dependency among labels. This is interesting because with no label dependence, BR is not the best method for F-measure and Jaccard distance. Also, CLR becomes very similar to BR at high levels of difficulty in all measures. This occurs in special when ϕ is sufficiently high because, for all $1 \leq i \leq m$, the probability $\mathbf{P}(Y_i = 1)$ approximates to $\frac{1}{2}$. For any $1 \leq i \leq m$, if $\mathbf{P}(Y_i) = \mathbf{P}(Y_j)$, $\forall j$, then CLR prediction at (2.6) becomes equal to BR prediction at (2.4). In order to prove that, assume that $\mathbf{P}(Y_i) = \mathbf{P}(Y_j)$, $\forall j$ for some label i. Therefore, $\mathbf{P}(Y_i = 1, Y_j = 0) = \mathbf{P}(Y_i = 0, Y_j = 1)$. From (2.6), it can be concluded that the prediction of CLR is defined by the expression

$$\sum_{j \neq i} \frac{\mathbf{P}(Y_i = 1, Y_j = 0)}{2 \cdot \mathbf{P}(Y_i = 1, Y_j = 0)} \; + \; \mathbf{P}(Y_i = 1) > \sum_{j=1}^{n} \mathbf{P}(Y_j = 0),$$

which is equivalent to

$$\sum_{j \neq i} \frac{1}{2} \; + \; \mathbf{P}(Y_i = 1) > n(1 - \mathbf{P}(Y_i = 1)),$$

and finally

$$\mathbf{P}(Y_i = 1) > \frac{1}{2}.$$

Chapter Conclusion

In the face of discussions and results presented in this chapter, one may conclude that our proposed method can aid in elaborating and testing new heuristics for optimizing predictions in multi-label classification. This is only successfully achieved because the proposed experimental framework has accomplished two crucial aspects of experimental setups of this kind in the context of loss minimization in MLC (DEMBCZYŃSKI et al., 2012):

- Low number of assumptions with respect to the simulation of the label distribution.
- Two important parameters can be easily controlled, the label dependence level and the difficulty of the problem.

To the best of our knowledge, no work has accomplished them simultaneously in its experimental setup.

The experimental study in Section 5.3 was carried out mainly to show the potential of using the proposed framework. Despite its simplicity, the study showed to be consistent with current works in multi-label classification, especially within Pereira et al. (2018), where authors draw similar conclusions with respect to which multi-label evaluation measures have strong relationships among each other.

More importantly, it was highlighted as useful as it was discovered that Jaccard distance optimizer and F-measure optimizer are both very related and the latter can be used as an approximation to the former. However, there is still no proof that CC is an approximation for optimizing subset 0/1 loss, the results in Section 5.2 reinforces existent arguments and shows a strong evidence towards the veracity of this statement. Moreover, it show that some methods have a strong correlation with each other, BR with CLR, Jaccard distance optimizer with F-measure optimizer, and CC with PCC. This is an important conclusion that may help researchers when comparing methods. Additionally, it was shown that label dependence is not the only relevant variable in multi-label classification problems. As the experiments have highlighted, the difficulty of the problem can either make multi-label methods better or worse than others.

Future works may address other practical usage of the framework, like investigating the impacts of changing the number of labels and/or the number of simulations. Furthermore, investigating if the framework may be used to detect the cases in which a multi-label classifier optimizes a metric. If so, further investigate if it can be done in a feasible time or it does require an extremely high number of simulations to obtain the solution. The framework can also be used to study the effects of changing the parameters of multi-label methods, such as the k in RAKEL, and the chain order of CC. And finally, a future work may consider the effectiveness of evaluating the multi-label method based on a stacking behaviour (like DBR) inside the framework.

This evaluation proposal is clearly not perfect and comes with some limitations. One obvious limitation is the assumption of a Beta distribution that generates parameters for other Beta distributions and then finally generates the probability tree. This is much better than assuming that all nodes representing the marginal distribution of a label follows identically the same Beta distribution (SULAIMAN et al., 1999).

Another limitation is the maximum number of labels to be considered, which cannot be larger than 20 labels in practice due to the exponential growth of the number of possible labellings. This maximum could be significantly increased if special characteristics for specific metrics and multi-label classifiers are exploited in such a way that the expected loss does not need to be computed for each one of the 2^m values of the joint distribution. In this case, not all values of the joint distribution need to be generated. The simplest example in which this occurs is evaluating CC with respect to subset 0/1 loss, in which only n nodes of the probability tree needs to be used in order to evaluate the risk (see the end of Section 5.2).

Additionally, in order to include a multi-label method in the framework, one need to express its prediction by a function (or algorithm) of the joint probability distribution of the labels (ex: Equation (2.4)). It is not known if any adaptation-based multi-label method can be evaluated in the framework. One may conjecture that any multi-label adaptation method can be viewed as a transformation-based method. If this is true, the framework may be used. However, this is an open question hard to prove. Finally, the most relevant limitation is probably the assumption of no estimation errors and no systematic bias in the given joint label distribution. Future work may address these limitations.

6 Conclusions

This thesis presented three studies that tackle the subject of dependency among labels in distinct ways. In Chapter 3 it was shown that the dependence among labels makes the problem of optimizing the expected coverage a NP-HARD problem, even when assuming a restricted scenario where all instances have exactly two labels or even when assuming a pairwise low level of dependence among labels. It is concluded therefore that optimizing the expected coverage is not tractable for problems with large number of labels, unless some strong assumptions are made such as label independence. In Chapter 4 it was proven that CLR can have a poor performance on multiple metrics in particular families of label probability distributions, even when considering a very low level of dependency among labels. This only occurs due to the way CLR exploits dependencies among labels. This certainly shows that just exploiting dependencies among labels does not make a multi-label method good, in fact, it can make it worse in some scenarios. In Chapter 5 an experimental study was conducted where multi-label algorithms were compared against each other in a scenario where the label dependence is quantified and conveniently controlled. With respect to the label dependence, results suggest that the label dependence is the most relevant factor that makes predictions far from the optimal prediction. Each chapter has presented its own conclusions regarding the details on how the label dependence impacts multi-label methods, showing it is critical to give attention to label dependence when using/designing algorithms. With this said, it can finally be concluded that this thesis achieved its objective in presenting valuable information on the analysis of the dependencies among labels in the field of multi-label learning.

APPENDIX A – Appendices for Chapter 3

A.1 Proof of Lemma 3.1

Lemma 3.1. *Let $m = |R_\phi(x)|$. For any arbitrary integer $0 \leq x < n$ and for any constant value $c \in \mathbb{R}$, if w' is a weighting function where*

$$w'(u,v) = c, \quad \forall u, v \in R_\phi(x),$$

then

$$f_\phi(R_\phi(x), w') = c\binom{m}{3} + cx\binom{m-1}{2}. \tag{3.4}$$

See Appendix A.1. □

Proof Rewrite f as

$$f_\phi(R_\phi(x), w) = \sum_{i=x}^{n-2} i\; W_\phi(\phi^{-1}(i)),$$

where ϕ^{-1} is the inverse function of ϕ, that is, $\phi^{-1}(i)$ returns the vertex at rank i. Note that $m = n - x$. Therefore

$$\begin{aligned} f_\phi(R_\phi(x), w') \;&=\; \sum_{i=x}^{n-2} ic(n-i-1) \\ &= c\sum_{i=0}^{n-2-x} (i+x)(n-i-x-1) \\ &= c\sum_{i=0}^{m-2} i(m-i-1) \;+\; c\sum_{i=0}^{m-2} x(m-i-1) \\ &= c\binom{m}{3} + cx\binom{m}{2}. \end{aligned}$$

□

A.2 Proof of Lemma 3.2

Lemma 3.2. *Given a non-weighted graph $G = (V, E)$, let $G' = \mathcal{T}(G) = (V', w)$ and define Q as the set of vertices such that $V' = V \cup Q$ and $V \cap Q = \varnothing$. For an arbitrary solution (ranking) ϕ of Max 2-SL consider the partition $(A_\phi, B_\phi, C_\phi) = \mathcal{P}_\phi(V, Q)$. If there exists a vertex $v \in C_\phi$ in which ($\{u, v\} \in E$, $\forall u \in B_\phi$), then there exists a solution π in which $B_\pi = B_\phi \cup \{v\}$, $C_\pi = C_\phi - \{v\}$ and $f_\pi(V, w) \geq f_\phi(V, w)$.*

Proof By assumption there exists a non-empty set S_ϕ such that

$$S_\phi = \{v \in C_\phi \mid \{v,u\} \in E,\ \forall u \in B_\phi\}. \tag{A.1}$$

Define x as the vertex in S_ϕ with the highest rank:

$$\phi(x) \geq \phi(v),\ \forall v \in S_\phi.$$

If there exists a vertex $v \in C_\phi$ in which $\phi(v) > \phi(x)$, then swap their ranks. This increases the objective function because v is connected to a vertex of B_ϕ via an edge with negative weight. Let $Q' = \{v \in Q \mid \phi(v) > \phi(x)\}$. By defining a new ranking function π as

$$\pi(v) = \begin{cases} \phi(x) + |Q'|, & \text{if } v = x \\ \phi(v) - 1, & \text{if } v \in Q' \\ \phi(v), & \text{otherwise,} \end{cases}$$

it can be shown that $f_\pi(V', w) \geq f_\phi(V', w)$. Note that $\pi(v) = \phi(v)$ for all $v \in V$, except when $v = x$. Also note that $W_\pi(v) = W_\phi(v)$ for all $v \in V$. Therefore, the difference $f_\pi(V', w) - f_\phi(V', w)$ according to (3.3) can easily be calculated as follows

$$\begin{aligned} f_\pi(V', w) - f_\phi(V', w) =& \pi(x)W_\phi(x) \ - \ \phi(x)W_\phi(x) \\ =& \left(\pi(x) - \phi(x)\right) W_\phi(x) \\ =& |Q'|\ W_\phi(x). \end{aligned}$$

By assumption (A.1), x is connected to all vertices in B_ϕ, meaning that $\forall v (v \in B_\phi \rightarrow w(x,v) = 1)$, therefore $W_\phi(x) = |B_\phi|$. Consequently,

$$f_\pi(V', w) - f_\phi(V', w) = |Q'|\ W_\phi(x) \geq 0.$$

Note that $|C_\pi| < |C_\phi|$.

□

A.3 Proof of Lemma 3.3

Lemma 3.3. *Given a non-weighted graph $G = (V, E)$, let $G' = \mathcal{T}(G) = (V', w)$ and define Q as the set of vertices such that $V' = V \cup Q$ and $V \cap Q = \varnothing$. For an arbitrary solution (ranking) ϕ of Max 2-SL consider the partition $(A_\phi, B_\phi, C_\phi) = \mathcal{P}_\phi(V, Q)$. If there exists a vertex $v \in (B_\phi \cup C_\phi)$ in which $(\{u,v\} \notin E,\ \exists u \in B_\phi)$, then there exists a solution π in which $A_\pi = A_\phi \cup \{v\}$, $C_\pi = C_\phi - \{v\}$, $B_\pi = B_\phi - \{v\}$ and $f_\pi(V, w) \geq f_\phi(V, w)$.*

Proof The solution ϕ is decomposed into two cases:

1. When there exists $v \in C_\phi$ in which $(\{u,v\} \notin E,\ \exists u \in B_\phi)$;

2. When there exists $v \in B_\phi$ in which $(\{u, v\} \notin E,\ \exists u \in B_\phi)$;

For the first case, the proof is analogous to the proof of Lemma 3.2. Define a set $S_\phi = \{v \in C_\phi : \{v, u\} \notin E,\ \exists u \in B_\phi\}$, denoting x as the vertex in S_ϕ with the lowest rank and define the new ranking function π as:

$$\pi(v) = \begin{cases} \phi(x) - |Q'|, & \text{if } v = x \\ \phi(v) + 1, & \text{if } v \in Q' \\ \phi(v), & \text{otherwise,} \end{cases}$$

where $Q' = \{v \in Q : \phi(v) < \phi(x)\}$. The difference $f_\pi(V', w) - f_\phi(V', w)$ according to (3.3) can easily be calculated as follows

$$\begin{aligned} f_\pi(V', w) - f_\phi(V', w) =& \pi(x)W_\phi(x) \ - \ \phi(x)W_\phi(x) \\ =& \left(\pi(x) - \phi(x)\right) W_\phi(x) \\ =& - \ |Q'| \ W_\phi(x). \end{aligned}$$

By assumption, there exists $y \in X_R$ in which $w(x, y) = -n$, which implies that $W_\phi(x)$ is negative. Consequently,

$$f_\pi(V', w) - f_\phi(V', w) = -|Q'| \ W_\phi(x) \geq 0.$$

Now, suppose the second case is true and let $S = \{v \in B_\phi : \{v, u\} \notin E,\ \exists u \in B_\phi\}$. By assumption, S is a non-empty set. Name x as the vertex in S with the lowest rank, that is:

$$\phi(x) \leq \phi(v),\ \forall v \in S.$$

Let X_L be $\{v \in B_\phi : \phi(v) < \phi(x)\}$ and $X_R = \{v \in B_\phi : \phi(v) > \phi(x)\}$. By defining a new ranking function π as

$$\pi(v) = \begin{cases} \phi(x) - |X_L| - |Q|, & \text{if } v = x \\ \phi(v) + 1, & \text{if } v \in X_L \ \text{ or } \ v \in Q \\ \phi(v), & \text{otherwise,} \end{cases}$$

it can be shown that $f_\pi(V', w) \geq f_\phi(V', w)$. It is true that

$$w(x, v) = 1,\ \forall v \in X_L.$$

Otherwise, there would be a vertex $v \in S$ with a lower rank than x. Note that the following statements are true:

$$\pi(v) = \phi(v) \ \text{ and } \ W_\pi(v) = W_\phi(v), \quad \forall v \in (A_\phi \cup X_R).$$

$$\begin{aligned} W_\pi(v) \ &= \ W_\phi(v) - 1, \quad \forall v \in X_L. \\ W_\pi(x) \ &= \ W_\phi(x) + |X_L|. \end{aligned}$$

Having the above notes in mind, the difference $f_\pi(V',w) - f_\phi(V',w)$ is calculated as

$$\begin{aligned}
& f_\pi(V',w) - f_\phi(V',w) \\
&= \left[\pi(x)W_\pi(x) + \sum_{v\in X_L} \pi(v)W_\pi(v)\right] \\
&\quad - \left[\phi(x)W_\phi(x) + \sum_{v\in X_L} \phi(v)W_\phi(v)\right] \\
&= \left[\pi(x)W_\phi(x) + \pi(x)|X_L| + \sum_{v\in X_L} (\phi(v)+1)\,(W_\phi(v)-1)\right] \\
&\quad - \left[\phi(x)W_\phi(x) + \sum_{v\in X_L} \phi(v)W_\phi(v)\right] \\
&= (\pi(x)-\phi(x))\,W_\phi(x) \ + \ \pi(x)|X_L| \\
&\quad + \sum_{v\in X_L} (-\phi(v)+W_\phi(v)-1) \\
&= -\,(|X_L|+|Q|)\,W_\phi(x) \ + \ \pi(x)|X_L| \\
&\quad + \sum_{v\in X_L} (-\phi(v)+W_\phi(v)-1)\,.
\end{aligned}$$

By assumption, there exists $y \in X_R$ in which $w(x,y) = -n$, which implies that

$$\begin{aligned}
W_\phi(x) = \sum_{v\in X_R} w(x,v) &= \sum_{v\in X_R: v\neq y} w(x,v) \ - \ n \\
&\leq \ |X_R| - 1 - n \ \leq \ -|X_L| - 1.
\end{aligned}$$

Therefore,

$$\begin{aligned}
& f_\pi(V',w) - f_\phi(V',w) \\
&\geq \ (|X_L|+|Q|)(|X_L|+1) + \pi(x)|X_L| - \sum_{v\in X_L} (\phi(v)+1) \\
&\geq \ (|X_L|+|Q|)(|X_L|+1) + (\phi(x) - |X_L| + |Q|)|X_L| \\
&\qquad - \ \phi(x)|X_L| \\
&= \ (|X_L|+|Q|)(|X_L|+1) - (|X_L|+|Q|)|X_L| \\
&= \ |X_L| + |Q|.
\end{aligned}$$

□

A.4 Proof of Lemma 3.4

Lemma 3.4. *Given a non-weighted graph $G = (V,E)$, let $G' = \mathcal{T}(G) = (V',w)$, and define Q as the set of vertices such that $V' = V \cup Q$ and $V \cap Q = \varnothing$. For an arbitrary solution (ranking) ϕ of Max 2-SL consider the partition $(A_\phi, B_\phi, C_\phi) = \mathcal{P}_\phi(V,Q)$. There exists a ranking π in which $f_\pi(V,w) \geq f_\phi(V,w)$, $C_\pi = \varnothing$ and B_π is a clique in G.*

Proof Suppose that $C_\phi \neq \varnothing$ or B_ϕ is not a clique in G. Solution ϕ satisfies at least one of the following conditions:

- There exists a vertex $v \in C_\phi$ in which $(\{u, v\} \in E,\ \forall u \in B_\phi)$;
- There exists a vertex $v \in (B_\phi \cup C_\phi)$ in which $(\{u, v\} \notin E,\ \exists u \in B_\phi)$.

In either of the above conditions, we can show by using Lemma 3.2 and Lemma 3.3 that there exists a new solution π such that $|B_\pi \cup C_\pi| < |B_\phi \cup C_\phi|$ and $f_\pi(V, w) \geq f_\phi(V, w)$. If $C_\pi \neq \varnothing$ or B_π is not a clique, we can recursively use the same argument used for ϕ to find new solutions until finally obtaining solution π^* such that $C_{\pi^*} = \varnothing$ and B_{π^*} is a clique in G. □

APPENDIX B – Appendices for Chapter 4

B.1 Proof of Proposition 4.1

Proposition 4.1. *When considering distribution $\hat{\mathbf{P}}_m$, CLR predicts ones for the first m labels and zeroes for the other labels, i.e, $\sum_{i=1}^{m} h_i^{clr} = \sum_{i=1}^{n} h_i^{clr} = m$.*

It will be shown that $\mathbf{h}^{\text{clr}}$ satisfies Inequality (2.6) for $1 \leq i \leq m$ on distribution $\hat{\mathbf{P}}_m$ if and only if $1 \leq i \leq m$. Firstly, it will be shown that (2.6) is not satisfied for $i > m$, that is

$$\sum_{j=1}^{n} f(\hat{\mathbf{P}}_m, i, j) + \hat{\mathbf{P}}_m^{(i)} < \sum_{j=1}^{n} \left(1 - \hat{\mathbf{P}}_m^{(j)}\right), \text{ for all } i > m. \tag{B.1}$$

Knowing that

$$\hat{\mathbf{P}}_m^{(i)} = \begin{cases} \hat{\mathbf{P}}_m(1_n) + \hat{\mathbf{P}}_m(\mathbf{y}^{(i)}) = \frac{m+1}{2(n+1)} + \epsilon, & \text{for } i \leq m, \\ \hat{\mathbf{P}}_m(1_n) = \frac{m+1}{2(n+1)}, & \text{for } i > m, \end{cases} \tag{B.2}$$

the right-hand side of (B.1) is equivalent to:

$$\sum_{j=1}^{n} \left(1 - \hat{\mathbf{P}}_m^{(j)}\right) = n - n \cdot \frac{m+1}{2(n+1)} - m\epsilon. \tag{B.3}$$

For the left-hand side of (B.1), and for $1 \leq j \leq m < i \leq n$, it can be observed that

$$\begin{aligned} f(\hat{\mathbf{P}}_m, i, j) &= \frac{\hat{\mathbf{P}}_m(Y_i = 1, Y_j = 0)}{\hat{\mathbf{P}}_m(Y_i = 1, Y_j = 0) + \hat{\mathbf{P}}_m(Y_i = 0, Y_j = 1)} \\ &= \frac{0}{0 + \hat{\mathbf{P}}_m(\mathbf{y}^{(j)})} = \frac{0}{\epsilon} = 0, \qquad \text{for } j \leq m < i. \end{aligned} \tag{B.4}$$

Using (B.4) and (B.3), (B.1) is equivalent to

$$\sum_{j=m+1}^{n} f(\hat{\mathbf{P}}_m, i, j) + \hat{\mathbf{P}}_m^{(i)} < n - n \cdot \frac{m+1}{2(n+1)} - m\epsilon, \quad \text{for all } i > m.$$

The last inequality is always satisfied, even if the left-hand side assumes an upper bound of $\sum_{j=m+1}^{n} f(\hat{\mathbf{P}}_m, i, j) \leq \sum_{j=m+1: j \neq i}^{n} 1 = n - m - 1$:

$$\begin{aligned} n - m - 1 + \frac{m+1}{2(n+1)} < n - n\frac{m+1}{2(n+1)} - m\epsilon &\iff -m - 1 < -(n+1)\frac{m+1}{2(n+1)} - m\epsilon \\ &\iff 2m + 2 > m + 1 + 2m\epsilon \\ &\iff m + 1 > 2m\epsilon. \end{aligned}$$

The last inequality is satisfied for a sufficiently small ϵ. This concludes the proof for $i > m$.

Now consider $i \leq m$. Let us show that

$$\sum_{j=1}^{n} f(\hat{\mathbf{P}}_m, i, j) + \hat{\mathbf{P}}_m^{(i)} > \sum_{j=1}^{n} \left(1 - \hat{\mathbf{P}}_m^{(i)}\right). \tag{B.5}$$

Firstly, note that, for if $1 \leq i \leq m < j \leq n \longrightarrow f(\hat{\mathbf{P}}_m, i, j) = \frac{\hat{\mathbf{P}}_m(\mathbf{y}^{(i)})}{\hat{\mathbf{P}}_m(\mathbf{y}^{(i)}) + \hat{\mathbf{P}}_m(\mathbf{y}^{(j)})} = \frac{\epsilon}{\epsilon+0} = 1$. Moreover, note that $f(\hat{\mathbf{P}}_m, i, j) = \frac{1}{2}$ for any $1 \leq i \leq m$, $1 \leq j \leq m$ and $i \neq j$. Therefore, Inequality (B.5) is equivalent to

$$\sum_{j=1:j\neq i}^{m} \frac{1}{2} + \sum_{j=m+1}^{n} 1 \; + \; \hat{\mathbf{P}}_m^{(i)} > \sum_{j=1}^{n} \left(1 - \hat{\mathbf{P}}_m^{(i)}\right),$$

and then

$$\frac{m-1}{2} + (n-m) + \hat{\mathbf{P}}_m^{(i)} > \sum_{j=1}^{n} \left(1 - \hat{\mathbf{P}}_m^{(i)}\right).$$

From (B.2), it has that $\hat{\mathbf{P}}_m^{(i)} = \frac{m+1}{2(n+1)} + \epsilon$ for all $i \leq m$, so the above inequality is equivalent to

$$\frac{m-1}{2} + n - m + \frac{m+1}{2(n+1)} + \epsilon > \sum_{j=1}^{n} \left(1 - \frac{m+1}{2(n+1)}\right),$$

and then simplifying

$$2n - m - 1 + \frac{m+1}{n+1} + \epsilon > 2n - n \cdot \frac{m+1}{n+1},$$

and again

$$-m - 1 + (n+1)\frac{m+1}{(n+1)} + \epsilon > 0,$$

and finally

$$\epsilon > 0,$$

which is, by definition, always true. □

B.2 Proof of Theorem 4.1

Theorem 4.1. *The following upper bound holds for the regret with respect to Hamming loss:*

$$\sup_{\mathbf{P} \in \mathcal{P}_n} \left(r_H(\mathbf{h}^{clr})\right) = \begin{cases} \frac{n}{4(n+1)}, & \text{if } n \text{ is even} \\ \frac{n-1}{4n}, & \text{if } n \text{ is odd,} \end{cases}$$

where $\mathcal{P}_n$ denotes the set of all distributions over n labels such that $\mathbf{P}^{(i)} \leq \frac{1}{2}$ for all i. See Appendix B.2. □

For an arbitrary distribution $\mathbf{P}$ of $\mathbf{Y}$, denote $\mathbf{y}^*$ as the optimal expected Hamming loss for $\mathbf{P}$. The risk of an arbitrary labelling $\hat{\mathbf{y}}$ with respect to Hamming loss can be written as

$$R_H(\hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} (1 - \mathbf{P}(Y_i = \hat{y}_i)), \tag{B.6}$$

which can be derived from the definition:

$$\begin{aligned}
R_H(\hat{\mathbf{y}}) &= \sum_{\mathbf{y}\in\mathcal{Y}} L_H(\mathbf{y},\hat{\mathbf{y}})\cdot \mathbf{P}(\mathbf{Y}=\mathbf{y}) \\
&= \sum_{\mathbf{y}\in\mathcal{Y}} \left(\frac{1}{n}\sum_{i=1}^{n} [\![y_i \neq \hat{y}_i]\!]\right)\cdot \mathbf{P}(\mathbf{Y}=\mathbf{y}) \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{\mathbf{y}\in\mathcal{Y}} [\![y_i \neq \hat{y}_i]\!]\cdot \mathbf{P}(\mathbf{Y}=\mathbf{y}) \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{\mathbf{y}\in\mathcal{Y}:y_i\neq\hat{y}_i} \mathbf{P}(\mathbf{Y}=\mathbf{y}) \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbf{P}(Y_i\neq\hat{y}_i) = \frac{1}{n}\sum_{i=1}^{n}(1-\mathbf{P}(Y_i=\hat{y}_i)).
\end{aligned}$$

The regret with respect to Hamming loss can be expressed as

$$r_H(\hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i^*)(1-2\mathbf{P}^{(i)}), \tag{B.7}$$

by using the definition of regret and (B.6):

$$\begin{aligned}
r_H(\hat{\mathbf{y}}) &= R_H(\hat{\mathbf{y}}) - R_H(\mathbf{y}^*) \\
&= \frac{1}{n}\sum_{i=1}^{n}(1-\mathbf{P}(Y_i=\hat{y}_i)) - \frac{1}{n}\sum_{i=1}^{n}(1-\mathbf{P}(Y_i=y_i^*)) \\
&= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{P}(Y_i=y_i^*) - \mathbf{P}(Y_i=\hat{y}_i)).
\end{aligned}$$

Note that

$$\mathbf{P}(Y_i=y_i^*) - \mathbf{P}(Y_i=\hat{y}_i) = \begin{cases} 0, & \text{if } y_i^*=\hat{y}_i, \\ \mathbf{P}(Y_i=0)-\mathbf{P}(Y_i=1), & \text{if } y_i^*=1 \text{ and } \hat{y}_i=0, \\ \mathbf{P}(Y_i=1)-\mathbf{P}(Y_i=0), & \text{if } y_i^*=0 \text{ and } \hat{y}_i=1. \end{cases}$$

Therefore

$$\begin{aligned}
r_H(\hat{\mathbf{y}}) &= \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i^*)(\mathbf{P}(Y_i=0)-\mathbf{P}(Y_i=1)) \\
&= \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i^*)(1-2\mathbf{P}^{(i)}).
\end{aligned}$$

Said that, define $A = \{i : y_i^* = 0 \wedge h_i^{\text{clr}} = 1\}$, i.e the set of all false positive labels, and $a = |A|$. From Equation (2.13), it is easy to see that $\mathbf{y}^* = 0_n$ for all distributions in $\mathcal{P}_n$, so

$$\begin{aligned}
r_H(\mathbf{h}^{\text{clr}}) &= \frac{1}{n}\sum_{i=1}^{n} h_i^{\text{clr}}(1-2\mathbf{P}^{(i)}) \\
&= \frac{1}{n}\left(a - 2\sum_{i\in A}\mathbf{P}^{(i)}\right)
\end{aligned} \tag{B.8}$$

The regret with respect to hamming loss $r_H(\mathbf{h}^{\mathrm{clr}})$ can be expressed as following

$$\begin{aligned} r_H(\mathbf{h}^{\mathrm{clr}}) &= \frac{1}{n}\sum_{i=1}^{n}(h_i^{\mathrm{clr}} - y_i^*)(1 - 2\mathbf{P}^{(i)}) \\ &= \frac{1}{n}\left(a - b + 2\sum_{i\in B}\mathbf{P}^{(i)} - 2\sum_{i\in A}\mathbf{P}^{(i)}\right). \end{aligned}$$

It is easy to see that $\mathbf{y}^* = 0_n$ for all distributions in $\mathcal{P}_n$, so $B = \varnothing$ which leads to

$$r_H(\mathbf{h}^{\mathrm{clr}}) = \frac{1}{n}\left(a - 2\sum_{i\in A}\mathbf{P}^{(i)}\right)$$

In the next steps, we will find a lower bound for $\sum_{i\in A}\mathbf{P}^{(i)}$, consequently giving an upper bound for $r_H(\mathbf{h}^{\mathrm{clr}})$. Summing the scores $\sum_{i\in A} s_i$, defined in (2.9), results in:

$$\begin{aligned} \sum_{i\in A} s_i &= \sum_{i\in A}\sum_{j=1}^{n} f(\mathbf{P},i,j) + \sum_{i\in A}\mathbf{P}^{(i)} \\ &= \sum_{i\in A}\sum_{j\in A} f(\mathbf{P},i,j) + \sum_{i\in A}\sum_{j\notin A} f(\mathbf{P},i,j) + \sum_{i\in A}\mathbf{P}^{(i)} \end{aligned}$$

Knowing that $f(\mathbf{P},i,j) + f(\mathbf{P},j,i) = 1$ for any $i \neq j$, we have that $\sum_{i\in A}\sum_{j\in A} f(\mathbf{P},i,j) = \frac{a(a-1)}{2}$, therefore

$$\sum_{i\in A} s_i = \frac{a(a-1)}{2} + \sum_{i\in A}\sum_{j\notin A} f(\mathbf{P},i,j) + \sum_{i\in A}\mathbf{P}^{(i)}. \tag{B.9}$$

Using the upper bound $f(\mathbf{P},i,j) \leq 1 - \mathbf{P}(Y_i = 0, Y_j = 1)$ for any i and j, it can be shown that

$$\begin{aligned} \sum_{i\in A}\sum_{j\notin A} f(\mathbf{P},i,j) &\leq \sum_{i\in A}\sum_{j\notin A}(1 - \mathbf{P}(Y_i = 0, Y_j = 1)) \\ &= \sum_{i\in A}\sum_{j\notin A}(1 + \mathbf{P}(Y_i = 1, Y_j = 1) - \mathbf{P}^{(j)}) \\ &\leq \sum_{i\in A}\sum_{j\notin A}(1 + \mathbf{P}^{(i)} - \mathbf{P}^{(j)}) \\ &= a(n-a) + \sum_{i\in A}\sum_{j\notin A}(\mathbf{P}^{(i)} - \mathbf{P}^{(j)}) \\ &= a(n-a) + (n-a)\sum_{i\in A}\mathbf{P}^{(i)} \;-\; a\sum_{j\notin A}\mathbf{P}^{(j)}. \end{aligned} \tag{B.10}$$

Using the upper bound at (B.10) on Equation (B.9):

$$\begin{aligned} \sum_{i\in A} s_i &\leq \frac{a(a-1)}{2} + a(n-a) + (n-a)\sum_{i\in A}\mathbf{P}^{(i)} \;-\; a\sum_{j\notin A}\mathbf{P}^{(j)} + \sum_{i\in A}\mathbf{P}^{(i)} \\ &= \frac{a(a-1)}{2} + a(n-a) + (n-a+1)\sum_{i\in A}\mathbf{P}^{(i)} \;-\; a\sum_{j\notin A}\mathbf{P}^{(j)} \\ &= -\frac{a(a+1)}{2} + an + (n-a+1)\sum_{i\in A}\mathbf{P}^{(i)} \;-\; a\sum_{j\notin A}\mathbf{P}^{(j)}. \end{aligned} \tag{B.11}$$

By definition of CLR, it is true that $\sum_{i\in A} s_i \geq a\left(n - \sum_{j=1}^{n} \mathbf{P}^{(j)}\right)$, when summing which is the sum of all conditions associated to false positive labels. Applying the upper bound in (B.11) on it:

$$-\frac{a(a+1)}{2} + an + (n-a+1)\sum_{i\in A}\mathbf{P}^{(i)} \ - \ a\sum_{j\notin A}\mathbf{P}^{(j)} \geq a\left(n - \sum_{j=1}^{n}\mathbf{P}^{(j)}\right).$$

Note that an is present on both sides, so it can be simplified to

$$-\frac{a(a+1)}{2} + (n-a+1)\sum_{i\in A}\mathbf{P}^{(i)} \ - \ a\sum_{j\notin A}\mathbf{P}^{(j)} \geq -a\sum_{j=1}^{n}\mathbf{P}^{(j)}.$$

Also note that $\sum_{j=1}^{n}\mathbf{P}^{(j)} = \sum_{j\in A}^{n}\mathbf{P}^{(j)} + \sum_{j\notin A}^{n}\mathbf{P}^{(j)}$, so the inequality is again simplified to

$$-\frac{a(a+1)}{2} + (n-a+1)\sum_{i\in A}\mathbf{P}^{(i)} \geq -a\sum_{j\in A}^{n}\mathbf{P}^{(j)},$$

which is equivalent to

$$(n+1)\sum_{i\in A}\mathbf{P}^{(i)} \geq \frac{a(a+1)}{2},$$

and finally

$$\sum_{i\in A}\mathbf{P}^{(i)} \geq \frac{a(a+1)}{2(n+1)}.$$

Using this last lower bound for $\sum_{i\in A}\mathbf{P}^{(i)}$ at Inequality (B.8), it can be derived an upper bound for $r_H(\mathbf{h}^{\text{clr}})$:

$$\begin{aligned}
r_H(\mathbf{h}^{\text{clr}}) &\leq \frac{1}{n}\left(a - 2\sum_{i\in A}\mathbf{P}^{(i)}\right)\\
&\leq \frac{1}{n}\left(a - 2\frac{a(a+1)}{2(n+1)}\right)\\
&= \frac{a}{n}\left(1 - \frac{a+1}{n+1}\right)\\
&= \frac{a}{n}\left(\frac{n-1+a+1}{n+1}\right)\\
&= \frac{a(n-a)}{n(n+1)},
\end{aligned}$$

which is a quadratic polynomial with respect to a that clearly has a maximum when $a = \frac{n}{2}$, if n is even. Therefore

$$r_H(\mathbf{h}^{\text{clr}}) \leq \frac{n}{4(n+1)}.$$

If n is odd, the maximum is given when $a = \frac{n-1}{2}$ or $a = \frac{n+1}{2}$.

To show that this bound is tight, it just needs to be shown the existence of a distribution that yields a regret arbitrarily as close to the value above. Distribution $\hat{\mathbf{P}}_m$ for $m = \frac{n}{2}$ satisfies this condition. Given that $\hat{\mathbf{P}}_{n/2}^{(i)} = \frac{n+2}{4(n+1)} + \epsilon < \frac{1}{2}$ for any i, the optimal labelling for Hamming loss is 0_n. Therefore, the value of $r_H(\mathbf{h}^{\text{clr}})$ on distribution $\hat{\mathbf{P}}_{n/2}$ is given by

$$\begin{aligned} r_H(\mathbf{h}^{\text{clr}}) =& \frac{1}{n}\sum_{i\in A}\left(1-2\hat{\mathbf{P}}^{(i)}_{n/2}\right) \\ =& \frac{1}{n}\sum_{i\in A}\left(1-\frac{n+2}{2(n+1)}-\epsilon\right), \end{aligned}$$

and then, using Proposition 4.1,

$$\begin{aligned} r_H(\mathbf{h}^{\text{clr}}) =& \frac{n}{2n}\left(1-\frac{n+2}{2(n+1)}-\epsilon\right) \\ =& \frac{1}{2}\left(\frac{2n+2-n-2}{2(n+1)}-\epsilon\right) \\ =& \frac{n}{4(n+1)}-\frac{\epsilon}{2}. \end{aligned}$$

□

B.3 Proof of Theorem 4.3

Theorem 4.3. *The following lower bound holds for the regret with respect to Jaccard distance:*

$$\sup_{\mathbf{P}} r_J(\mathbf{h}^{clr}) \geq 1-\frac{1}{n}.$$

Consider distribution $\hat{\mathbf{P}}_1$ and note that there are only three labellings with a non-null probability: 0_n, 1_n and $\mathbf{y}^{(1)}$. From Proposition 4.1, it has that $\mathbf{h}^{\text{clr}} = \mathbf{y}^{(1)}$ for distribution $\hat{\mathbf{P}}_1$. Given that the loss $L_J(1_n, \mathbf{h}^{\text{clr}})$ can be calculated as the following

$$L_J(1_n,\mathbf{h}^{\text{clr}}) = L_J(1_n,\mathbf{y}^{(1)}) = 1-\frac{\sum_{i=1}^n h_i^{\text{clr}}}{n+\sum_{i=1}^n h_i^{\text{clr}}-\sum_{i=1}^n h_i^{\text{clr}}} = 1-\frac{1}{n},$$

the risk of CLR is

$$\begin{aligned} R_J(\mathbf{h}^{\text{clr}}) = R_J(\mathbf{y}^{(1)}) &= \sum_{\mathbf{y}\in\{0_n,1_n,\mathbf{y}^{(1)}\}} L_J(\mathbf{y},\mathbf{y}^{(1)})\hat{\mathbf{P}}_1(\mathbf{y}) \\ &= \underbrace{L_J(0_n,\mathbf{y}^{(1)})}_{1}\ \underbrace{\hat{\mathbf{P}}_1(0_n)}_{1-1/(n+1)-\epsilon} + \underbrace{L_J(1_n,\mathbf{y}^{(1)})}_{1-1/n}\,\underbrace{\hat{\mathbf{P}}_1(1_n)}_{1/(n+1)} + \underbrace{L_J(\mathbf{y}^{(1)},\mathbf{y}^{(1)})}_{0}\,\hat{\mathbf{P}}_1(\mathbf{y}^{(1)}) \\ &= 1-\frac{1}{n+1}-\epsilon+\frac{1}{n+1}-\frac{1}{n(n+1)} \\ &= 1-\epsilon-\frac{1}{n(n+1)}. \end{aligned}$$

The risk of the optimal solution $\mathbf{y}^*$ is upper bounded by

$$\begin{aligned} R_J(\mathbf{y}^*) \le R_J(0_n) &= \sum_{\mathbf{y}\in\{0_n,1_n,\mathbf{y}^{(1)}\}} L_J(\mathbf{y},0_n)\hat{\mathbf{P}}_1(\mathbf{y}) \\ &= \underbrace{L_J(0_n,0_n)}_{0}\hat{\mathbf{P}}_1(0_n) + \underbrace{L_J(1_n,0_n)}_{1}\underbrace{\hat{\mathbf{P}}_1(1_n)}_{1/(n+1)} + \underbrace{L_J(\mathbf{y}^{(1)},0_n)}_{\le 1}\underbrace{\hat{\mathbf{P}}_1(\mathbf{y}^{(1)})}_{\epsilon} \\ &\le \frac{1}{n+1} + \epsilon. \end{aligned}$$

The regret is lower bounded by

$$\begin{aligned} r_J(\mathbf{h}^{\text{clr}}) &\ge \overbrace{1-\epsilon-\frac{1}{n(n+1)}}^{R_J(\mathbf{h}^{\text{clr}})} - \overbrace{\left(\frac{1}{n+1}+\epsilon\right)}^{R_J(0_n)} \\ &= 1-2\epsilon-\frac{n+1}{n(n+1)} \\ &= 1-2\epsilon-\frac{1}{n}. \end{aligned}$$

The value of ϵ can be made arbitrarily small, so

$$\sup r_J(\mathbf{h}^{\text{clr}}) \ge 1-\frac{1}{n}.$$

□

B.4 Proof of Theorem 4.4

Theorem 4.4. *The following lower bound holds for the regret with respect to F-measure:*

$$\sup_{\mathbf{P}} r_F(\mathbf{h}^{clr}) \ge 1-\frac{n+3}{(n+1)^2}.$$

Consider distribution $\hat{\mathbf{P}}_1$ and note that there are only three labellings with a non-null probability: 0_n, 1_n and $\mathbf{y}^{(1)}$. This proof is very similar to Theorem 4.3. From Proposition 4.1, it has that $\mathbf{h}^{\text{clr}} = \mathbf{y}^{(1)}$ for distribution $\hat{\mathbf{P}}_1$. Given that the loss $L_F(1_n,\mathbf{h}^{\text{clr}})$ can be calculated as the following

$$L_F(1_n,\mathbf{h}^{\text{clr}}) = L_F(1_n,\mathbf{y}^{(1)}) = 1-\frac{2\sum_{i=1}^n h_i^{\text{clr}}}{n+\sum_{i=1}^n h_i^{\text{clr}}} = 1-\frac{2}{n+1}.$$

the risk of CLR is

$$\begin{aligned} R_F(\mathbf{h}^{\text{clr}}) = R_F(\mathbf{y}^{(1)}) &= \sum_{\mathbf{y}\in\{0_n,1_n,\mathbf{y}^{(1)}\}} L_F(\mathbf{y},\mathbf{y}^{(1)})\hat{\mathbf{P}}_1(\mathbf{y}) \\ &= \underbrace{L_F(0_n,\mathbf{y}^{(1)})}_{1}\underbrace{\hat{\mathbf{P}}_1(0_n)}_{1-1/(n+1)-\epsilon} + \underbrace{L_F(1_n,\mathbf{y}^{(1)})}_{1-2/(n+1)}\underbrace{\hat{\mathbf{P}}_1(1_n)}_{1/(n+1)} + \underbrace{L_F(\mathbf{y}^{(1)},\mathbf{y}^{(1)})}_{0}\hat{\mathbf{P}}_1(\mathbf{y}^{(1)}) \\ &= 1-\frac{1}{n+1}-\epsilon+\frac{1}{n+1}-\frac{2}{(n+1)^2} \\ &= 1-\epsilon-\frac{2}{(n+1)^2}. \end{aligned}$$

The risk of the optimal solution $\mathbf{y}^*$ is upper bounded by

$$\begin{aligned}
R_F(\mathbf{y}^*) \leq R_F(0_n) &= \sum_{\mathbf{y}\in\{0_n,1_n,\mathbf{y}^{(1)}\}} L_F(\mathbf{y},0_n)\hat{\mathbf{P}}_1(\mathbf{y}) \\
&= \underbrace{L_F(0_n,0_n)}_{0}\hat{\mathbf{P}}_1(0_n) + \underbrace{L_F(1_n,0_n)}_{1}\underbrace{\hat{\mathbf{P}}_1(1_n)}_{1/(n+1)} + \underbrace{L_F(\mathbf{y}^{(1)},0_n)}_{\leq 1}\underbrace{\hat{\mathbf{P}}_1(\mathbf{y}^{(1)})}_{\epsilon} \\
&\leq \frac{1}{n+1} + \epsilon.
\end{aligned}$$

The regret is lower bounded by

$$\begin{aligned}
r_F(\mathbf{h}^{\mathrm{clr}}) &\geq \overbrace{1-\epsilon-\frac{2}{(n+1)^2}}^{R_F(\mathbf{h}^{\mathrm{clr}})} - \overbrace{\left(\frac{1}{n+1}+\epsilon\right)}^{R_F(0_n)} \\
&= 1-2\epsilon-\frac{n+3}{(n+1)^2} \\
&= 1-2\epsilon-\frac{n+3}{(n+1)^2}.
\end{aligned}$$

The value of ϵ can be made arbitrarily small, so

$$\sup r_F(\mathbf{h}^{\mathrm{clr}}) \geq 1-\frac{n+3}{(n+1)^2}.$$

□

B.5 Proof of Theorem 4.5

Theorem 4.5. *For any n divisible by 4, the following lower bound holds for the regret with respect to normalized rank loss:*

$$\sup_{\mathbf{P}} r_{\hat{r}}(\mathbf{h}^{rpc}) \geq \frac{1}{6}.$$

The proof is given by showing that a specific probability label distribution $\widetilde{\mathbf{P}}$ gives a regret of exactly $\frac{1}{6}$ for any n divisible by 4. Before defining $\widetilde{\mathbf{P}}$, let three disjoint sets of labels, A, B and C be defined as following (note that we are using integers to represent labels):

$$\begin{aligned}
A &= \{i\in\mathbb{Z} \mid 1\leq i\leq \frac{n}{4}\}, \\
B &= \{i\in\mathbb{Z} \mid \frac{n}{4} < i\leq \frac{n}{2}\}, \\
C &= \{i\in\mathbb{Z} \mid \frac{n}{2} < i\leq n\}.
\end{aligned}$$

Distribution $\widetilde{\mathbf{P}}$ is defined as

$$\widetilde{\mathbf{P}}(\mathbf{y}) = \begin{cases} \frac{3}{4} - n \cdot \epsilon, & \text{if all labels in } A \text{ are positive and all other labels are negative,} \\ \frac{1}{4}, & \text{if all labels in } A \text{ are negative and all other labels are positive,} \\ 2\epsilon, & \text{if exactly one label in } A \text{ is positive and all other labels are negative} \\ 2\epsilon, & \text{if exactly one label in } B \text{ is positive and all other labels are negative} \\ 0, & \text{otherwise} \end{cases}$$

where ϵ is an arbitrary positive real number that is assumed to be "really close" to 0. The purpose of ϵ in $\widetilde{\mathbf{P}}$ is identical to the purpose of ϵ in distribution $\hat{\mathbf{P}}_m$, which is to avoid undefined value for $f(\mathbf{P}, i, j)$ when the numerator and denominator are both null and to make $f(\mathbf{P}, i, j)$ be convenient values such as 1 or $\frac{1}{2}$.

It will be shown that RPC prefers any label in B to any label in A while the optimizer for rank loss prefers labels in A to labels in B. Consider an arbitrary pair of labels (i, j) where $i \in A$ and $j \in B$. Let's check that RPC prefers label j to i by checking which score s_i or s_j is higher:

$$\sum_{k=1}^{n} f(\widetilde{\mathbf{P}}, j, k) - \sum_{k=1}^{n} f(\widetilde{\mathbf{P}}, i, k) > 0 \text{ ?} \tag{B.12}$$

If the difference above (s_j-s_i) is positive, then RPC prefers label j to label i. The distribution $\widetilde{\mathbf{P}}$ has so few non-null values that it is easy to check, for all $i \in A$, that:

$$f(\widetilde{\mathbf{P}}, i, k) = \begin{cases} \frac{3/4 - n\epsilon + 2\epsilon}{1-(n-4)\epsilon}, & \text{if } k \in B, \\ \frac{3/4 - n\epsilon + 2\epsilon}{1-(n-2)\epsilon}, & \text{if } k \in C, \\ \frac{1}{2}, & \text{if } k \in A \ \wedge \ k \neq i. \end{cases}$$

For all $j \in B$, it can be also checked that

$$f(\widetilde{\mathbf{P}}, j, k) = \begin{cases} \frac{1}{2}, & \text{if } k \in B \ \wedge \ k \neq j, \\ 1, & \text{if } k \in C, \\ \frac{2\epsilon + 1/4}{1-(n-4)\epsilon}, & \text{if } k \in A \end{cases}$$

Therefore, the score s_i is rewritten as:

$$s_i = \sum_{k=1}^{n} f(\widetilde{\mathbf{P}}, i, k) = \frac{|A| - 1}{2} + |B| \frac{3/4 - n\epsilon + 2\epsilon}{1 - (n-4)\epsilon} + |C| \frac{3/4 - n\epsilon + 2\epsilon}{1 - (n-2)\epsilon}$$

Analogously, the score s_j is rewritten as:

$$s_j = \sum_{k=1}^{n} f(\widetilde{\mathbf{P}}, j, k) = |A| \frac{2\epsilon + 1/4}{1 - (n-4)\epsilon} + \frac{|B| - 1}{2} + |C|.$$

Note that $|A| = |B|$ so $\frac{|B|-1}{2}$ cancels out with $\frac{|A|-1}{2}$ on the difference $s_j - s_i$. Therefore the difference can be simplified to:

$$s_j - s_i = \left(|A| \frac{2\epsilon + 1/4}{1 - (n-4)\epsilon} + |C| \right) - \left(|B| \frac{3/4 - n\epsilon + 2\epsilon}{1 - (n-4)\epsilon} + |C| \frac{3/4 - n\epsilon + 2\epsilon}{1 - (n-2)\epsilon} \right)$$

It will be used the fact Given that $\frac{3/4-n\epsilon+2\epsilon}{1-(n-2)\epsilon} \leq \frac{3}{4}$ and $\frac{2\epsilon+1/4}{1-(n-4)\epsilon} \geq 2\epsilon + \frac{1}{4}$, a lower bound for $s_j - s_i$ can be found: to decrease the difference $s_j - s_i$.

$$s_j - s_i \geq |A|\left(2\epsilon + \frac{1}{4}\right) + |C| - \left(\frac{3|B|}{4} + \frac{3|C|}{4}\right).$$

It will be shown that this lower bound is positive. It will be seen later that this difference remains positive. Given that $2|A| = |C| = 2|B|$, it follows that

$$\begin{aligned} s_j - s_i &\geq |A|\left(2\epsilon + \frac{1}{4}\right) + 2|A| - \frac{3}{4}\cdot 3|A| \\ &= |A| \cdot 2\epsilon. \end{aligned}$$

The value $|A| \cdot 2\epsilon$ is always positive since $\epsilon > 0$ by definition. Therefore, it can be concluded that RPC prefers any label $j \in B$ to any label $i \in A$.

Instead of calculating the regret of the prediction of RPC ($\mathbf{h}^{\mathrm{rpc}}$) on distribution $\widetilde{\mathbf{P}}$, let us calculate the regret of the same prediction $\mathbf{h}^{\mathrm{rpc}}$, but on a new distribution $\widetilde{\mathbf{P}}_0$, which is defined in the same way as $\widetilde{\mathbf{P}}$, but with ϵ being zero. It will be shown that $|r_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}_0) - r_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}})| \leq n\epsilon 2^{n+1}$, where we are now using the notation where the probability distribution is an explicit parameter of the regret to avoid any confusion later. Although this upper bound seems a bit high, it is a multiple of ϵ, which can be arbitrarily made small. So when ϵ tends to zero, this difference also tends to zero. Note that we do not use $\widetilde{\mathbf{P}}_0$ from the beginning, because RPC prediction on $\widetilde{\mathbf{P}}_0$ is undefined. Observe that these two distributions slightly differ: $|\widetilde{\mathbf{P}}_0(\mathbf{y}) - \widetilde{\mathbf{P}}(\mathbf{y})| \leq n\epsilon$ for all $\mathbf{y}$. For any arbitrary ranking $\mathbf{z}$,

$$\begin{aligned} R_{\hat{r}}(\mathbf{z}, \widetilde{\mathbf{P}}_0) - R_{\hat{r}}(\mathbf{z}, \widetilde{\mathbf{P}}) &= \sum_{\mathbf{y}} L_{\hat{r}}(\mathbf{y}, \mathbf{z}) \underbrace{\left(\widetilde{\mathbf{P}}_0(\mathbf{y}) - \widetilde{\mathbf{P}}(\mathbf{y})\right)}_{\leq n\epsilon} \\ &\leq n\epsilon \sum_{\mathbf{y}} L_{\hat{r}}(\mathbf{y}, \mathbf{z}) = n\epsilon 2^n. \end{aligned}$$

The difference $|r_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}_0) - r_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}})|$ can not differ by twice of the above amount, since the regret is the difference of two risks.

$$\begin{aligned} r_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}_0) - r_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}) &= R_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}_0) - R_{\hat{r}}(\mathbf{z}_0^*, \widetilde{\mathbf{P}}_0) - \left(R_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}) - R_{\hat{r}}(\mathbf{z}^*, \widetilde{\mathbf{P}})\right) \\ &\leq R_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}_0) - R_{\hat{r}}(\mathbf{z}_0^*, \widetilde{\mathbf{P}}_0) - \left(R_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}) - R_{\hat{r}}(\mathbf{z}_0^*, \widetilde{\mathbf{P}})\right) \\ &= \underbrace{R_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}_0) - R_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}})}_{\leq n\epsilon 2^n} + \underbrace{R_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}) - R_{\hat{r}}(\mathbf{z}_0^*, \widetilde{\mathbf{P}}_0)}_{\leq n\epsilon 2^n} \\ &\leq n\epsilon 2^{n+1}. \end{aligned}$$

This can be done similarly with $r_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}) - r_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}_0)$, so

$$|r_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}_0) - r_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}})| \leq n\epsilon 2^{n+1}. \tag{B.13}$$

To calculate the regret, it necessary to know what is the optimal solution for $\widetilde{\mathbf{P}}_0$. Observe that

$$s_{\mathbf{y}}(n - s_{\mathbf{y}}) = \frac{n}{4} \cdot \frac{3n}{4} = \frac{3n^2}{16}, \tag{B.14}$$

where $s_{\mathbf{y}} = \sum y_i$, for all $\mathbf{y}$ such that $\widetilde{\mathbf{P}}_0(\mathbf{y}) > 0$. Hence, the optimal solution for normalized rank loss in this distribution is exactly the same of rank loss, as observed in Equation (2.15). To show the optimizer for rank loss prefers labels in A to labels in B, it just has to be shown that $\widetilde{\mathbf{P}}(Y_i = 1) - \widetilde{\mathbf{P}}(Y_j = 1) > 0$, for all $i \in A$ and all $j \in B$:

$$\widetilde{\mathbf{P}}_0(Y_i = 1) - \widetilde{\mathbf{P}}_0(Y_j = 1) = \frac{3}{4} - \frac{1}{4} = \frac{1}{2}. \tag{B.15}$$

Hence, it can be concluded that the optimizer for rank loss prefers labels from A.

So RPC makes at least $|A| \cdot |B| = \frac{n^2}{16}$ misorder. The regret given by each of these mistakes, as defined in Equation (2.14), are all equal and given by $\widetilde{\mathbf{P}}_0(Y_i = 1) - \widetilde{\mathbf{P}}_0(Y_j = 1)$ for $i \in A$ and $j \in B$. From Equation (B.15), we have that $\widetilde{\mathbf{P}}_0(Y_i = 1) - \widetilde{\mathbf{P}}_0(Y_j = 1) = \frac{1}{2}$. From Equation (2.14), the regret $r_{\hat{R}}(\mathbf{h}^{RPC})$ on $\widetilde{\mathbf{P}}_0$ is given by multiplying the number of misorder $(\frac{n^2}{16})$ by $\frac{1}{2}$ and dividing by the constant normalization factor of Equation (B.14):

$$r_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}}_0) = \frac{n^2}{16} \cdot \frac{1}{2} \cdot \frac{16}{3n^2} = \frac{1}{6}.$$

From the equation above and from (B.13), the regret $r_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}})$ differs from 1/6 only by a multiple of ϵ. Since ϵ can be arbitrarily small, the supreme of $r_{\hat{r}}(\mathbf{h}^{\mathrm{rpc}}, \widetilde{\mathbf{P}})$ is at least $\frac{1}{6}$.

□

B.6 Proof of Proposition 4.2

Proposition 4.2. *For distribution $\bar{\mathbf{P}}$ of n labels, CLR will predict 0_n.*

It will be shown that $s_1 = s_2 < \sum_i(1 - \bar{\mathbf{P}}^{(i)})$ and $s_3 = s_4 = ... = s_n < \sum_i(1 - \bar{\mathbf{P}}^{(i)})$ (see (2.9)). Firstly, calculate $\sum_i(1 - \bar{\mathbf{P}}^{(i)})$. Knowing that

$$\bar{\mathbf{P}}^{(1)} = \underbrace{\bar{\mathbf{P}}(Y_1 = 1, Y_2 = 0)}_{1/2-2\epsilon} + \underbrace{\bar{\mathbf{P}}(Y_1 = 1, Y_2 = 1)}_{\epsilon} = \frac{1}{2} - \epsilon$$
$$\bar{\mathbf{P}}^{(2)} = \underbrace{\bar{\mathbf{P}}(Y_1 = 0, Y_2 = 1)}_{1/2-2\epsilon} + \underbrace{\bar{\mathbf{P}}(Y_1 = 1, Y_2 = 1)}_{\epsilon} = \frac{1}{2} - \epsilon,$$

it has that

$$\begin{aligned}\sum_{i=1}^{n}(1 - \bar{\mathbf{P}}^{(i)}) &= n - \underbrace{\bar{\mathbf{P}}^{(1)}}_{1/2-\epsilon} - \underbrace{\bar{\mathbf{P}}^{(2)}}_{1/2-\epsilon} - \sum_{i=3}^{n} \underbrace{\bar{\mathbf{P}}^{(i)}}_{\phi_n} \\ &= n - 1 + 2\epsilon - (n-2)\phi_n.\end{aligned} \tag{B.16}$$

Now, it will be shown that $s_1 \leq n - 1 - \epsilon < \sum_i (1 - \bar{\mathbf{P}}^{(i)})$. Before that, note that $\bar{\mathbf{P}}(Y_1 = 1, Y_2 = 0) = \bar{\mathbf{P}}(Y_1 = 0, Y_2 = 1)$, implying that $f(\bar{\mathbf{P}}, 1, 2) = f(\bar{\mathbf{P}}, 2, 1) = 1/2$. Analogously, for any pair of labels $i, j \geq 3$ and $i \neq j$, it has that $f(\bar{\mathbf{P}}, i, j) = f(\bar{\mathbf{P}}, j, i) = 1/2$. Said that, an upper bound for s_1 is

$$\begin{aligned} s_1 &= f(\bar{\mathbf{P}}, 1, 2) + \underbrace{\bar{\mathbf{P}}^{(1)}}_{1/2-\epsilon} + \sum_{j=3}^{n} f(\bar{\mathbf{P}}, 1, j) \\ &= \frac{1}{2} + \frac{1}{2} - \epsilon + \sum_{j=3}^{n} \underbrace{f(\bar{\mathbf{P}}, 1, j)}_{\leq 1} \\ &\leq 1 - \epsilon + n - 2 \;=\; n - 1 - \epsilon, \end{aligned}$$

and $n - 1 - \epsilon$ is lesser than $\sum_i (1 - \bar{\mathbf{P}}^{(i)})$, because their difference is negative:

$$\begin{aligned} (n-1-\epsilon) - \sum_i (1 - \bar{\mathbf{P}}^{(i)}) &= (n-1-\epsilon) - (n - 1 + 2\epsilon - (n-2)\phi_n) && \text{From Equation (B.16)} \\ &= -3\epsilon + (n-2)\phi_n < 0. && \text{By definition } \phi_n < \frac{\epsilon}{n}. \end{aligned}$$

It will be shown that $s_3 = s_4 = ... = s_n \leq \sum_i (1 - \bar{\mathbf{P}}^{(i)})$. An upper bound for s_3 is given by

$$\begin{aligned} s_3 &= f(\bar{\mathbf{P}}, 3, 1) + f(\bar{\mathbf{P}}, 3, 2) + \underbrace{\bar{\mathbf{P}}^{(3)}}_{\phi_n} + \sum_{j=4}^{n} \underbrace{f(\bar{\mathbf{P}}, 3, j)}_{1/2} \\ &= f(\bar{\mathbf{P}}, 3, 1) + f(\bar{\mathbf{P}}, 3, 2) + \phi_n + \frac{n-3}{2} \\ &\leq 2 + \phi_n + \frac{n-3}{2} \;=\; \phi_n + \frac{n+1}{2}. \end{aligned}$$

It is easy to see that $s_3 \leq \phi_n + \frac{n+1}{2} \leq n - 1 + 2\epsilon - (n-2)\phi_n$, for a sufficiently large n $(n \geq 3)$. □

B.7 Proof of Theorem 4.6

Theorem 4.6. *The following expression holds for the regret with respect to subset 0/1 loss*

$$r_s(\mathbf{h}^{clr}) = \left(\frac{1}{2} - 5\epsilon\right) \cdot (1 - \phi_n)^{n-2}, \; \textit{for distribution } \bar{\mathbf{P}},$$

and, consequently

$$\lim_{n \to \infty, \epsilon \to 0} r_s(\mathbf{h}^{clr}) = \frac{1}{2}.$$

Clearly, the mode of $\bar{\mathbf{P}}$ is either $\mathbf{y}^{(1)}$ or $\mathbf{y}^{(2)}$. In both cases, the risk is the same:

$$\begin{aligned} R_s(\mathbf{y}^{(1)}) = 1 - \bar{\mathbf{P}}(\mathbf{y}^{(1)}) &= 1 - \bar{\mathbf{P}}(1, 0) \cdot \left(1 - \bar{\mathbf{P}}^{(3)}\right)^{n-2} \\ &= 1 - \left(\frac{1}{2} - 2\epsilon\right) \cdot (1 - \phi_n)^{n-2}. \end{aligned}$$

The risk of CLR is given by

$$\begin{aligned} R_s(\mathbf{h}^{\text{clr}}) = R_s(0_n) &= 1 - \bar{\mathbf{P}}(0,0) \cdot \left(1 - \bar{\mathbf{P}}^{(3)}\right)^{n-2} && \text{From Proposition 4.2} \\ &= 1 - 3\epsilon \cdot (1-\phi_n)^{n-2}. \end{aligned}$$

And finally, the regret is

$$\begin{aligned} r_s(\mathbf{h}^{\text{clr}}) &= R_s(\mathbf{h}^{\text{clr}}) - R_s(\mathbf{y}^{(1)}) \\ &= \left(\frac{1}{2} - 2\epsilon\right) \cdot (1-\phi_n)^{n-2} - 3\epsilon \cdot (1-\phi_n)^{n-2} \\ &= \left(\frac{1}{2} - 5\epsilon\right) \cdot (1-\phi_n)^{n-2}. \end{aligned}$$

By definition, $\lim_{n\to\infty}(1-\phi_n)^{n-2} = 1$, so

$$\lim_{n\to\infty,\epsilon\to 0} r_s(\mathbf{h}^{\text{clr}}) = \frac{1}{2}.$$

□

B.8 Proof of Theorem 4.7

Theorem 4.7. *The following expression holds for the regret with respect to Jaccard distance*

$$\lim_{\epsilon\to 0} r_J(\mathbf{h}^{clr}) = \frac{1}{2}, \text{ for distribution } \bar{\mathbf{P}}.$$

Let A be a set of labellings of n labels defined as $A = \{0_n, \mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \mathbf{y}^{(1,2)}\}$, and $A' = \mathcal{Y}\backslash A$ its complement of labellings of n labels. Let the risk be expressed as the following

$$R_L(\hat{\mathbf{y}}) = \sum_{\mathbf{y}\in\mathcal{Y}} L(\mathbf{y},\hat{\mathbf{y}})\bar{\mathbf{P}}(\mathbf{y}) = \sum_{\mathbf{y}\in A} L(\mathbf{y},\hat{\mathbf{y}})\bar{\mathbf{P}}(\mathbf{y}) + \sum_{\mathbf{y}\in A'} L(\mathbf{y},\hat{\mathbf{y}})\bar{\mathbf{P}}(\mathbf{y}).$$

It will be shown that $\sum_{\mathbf{y}\in A'} L_J(\mathbf{y},\hat{\mathbf{y}})\bar{\mathbf{P}}(\mathbf{y}) \le \epsilon/2$:

$$\begin{aligned} \sum_{\mathbf{y}\in A'} L_J(\mathbf{y},\hat{\mathbf{y}})\bar{\mathbf{P}}(\mathbf{y}) &\le \sum_{\mathbf{y}\in A'} \bar{\mathbf{P}}(\mathbf{y}) = 1 - \sum_{\mathbf{y}\in A} \bar{\mathbf{P}}(\mathbf{y}) \\ &= 1 - \left(\bar{\mathbf{P}}(0_n) + \bar{\mathbf{P}}(\mathbf{y}^{(1)}) + \bar{\mathbf{P}}(\mathbf{y}^{(2)}) + \bar{\mathbf{P}}(\mathbf{y}^{(1,2)})\right) \\ &= 1 - \left(\underbrace{\bar{\mathbf{P}}(0,0) + \bar{\mathbf{P}}(1,0) + \bar{\mathbf{P}}(0,1) + \bar{\mathbf{P}}(1,1)}_{1}\right) \cdot \underbrace{\bar{\mathbf{P}}(Y_3 = 0)\cdots\bar{\mathbf{P}}(Y_n = 0)}_{(1-\phi_n)^{n-2}} \\ &= 1 - (1-\phi_n)^{n-2} \le (n-2)\phi_n, \end{aligned}$$

where the last inequality comes from the Bernoulli inequality. By definition of ϕ_n it has that $(n-2)\phi_n < \epsilon/2$, which can be made arbitrarily small. Therefore,

$$\sum_{\mathbf{y}\in A} L_J(\mathbf{y},\hat{\mathbf{y}})\bar{\mathbf{P}}(\mathbf{y}) \le R_J(\hat{\mathbf{y}}) \le \epsilon + \sum_{\mathbf{y}\in A} L_J(\mathbf{y},\hat{\mathbf{y}})\bar{\mathbf{P}}(\mathbf{y}),$$

which implies that

$$\lim_{\epsilon\to 0} R_J(\hat{\mathbf{y}}) = \sum_{\mathbf{y}\in A} L_J(\mathbf{y},\hat{\mathbf{y}})\bar{\mathbf{P}}(\mathbf{y}). \tag{B.17}$$

Our objective is to calculate $\lim_{\epsilon\to 0} r_J(\mathbf{h}^{\text{clr}}) = \lim_{\epsilon\to 0} R_J(\mathbf{h}^{\text{clr}}) - \lim_{\epsilon\to 0} R_J(\mathbf{y}^*)$. When ϵ tends to zero, ϕ_n tends to zero and $\bar{\mathbf{P}}$ will have only 2 non-null probabilities, $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)}$. Thus, calculating the $\lim_{\epsilon\to 0} r_J(\mathbf{h}^{\text{clr}})$ is easy since there will be only 2 non-null probabilities to sum up. Hence, Equation (B.17) can be reduced to

$$\lim_{\epsilon\to 0} R_J(\hat{\mathbf{y}}) = L_J(\mathbf{y}^{(1)},\hat{\mathbf{y}})\bar{\mathbf{P}}(\mathbf{y}^{(1)}) + L_J(\mathbf{y}^{(2)},\hat{\mathbf{y}})\bar{\mathbf{P}}(\mathbf{y}^{(2)}). \tag{B.18}$$

Firstly, let us determine the optimal risk. An optimal solution for Jaccard distance on $\bar{\mathbf{P}}$ is clearly either $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}$ or $\mathbf{y}^{(1,2)}$. This can be easily solved by checking all three values.

$$\begin{aligned}\lim_{\epsilon\to 0} R_J(\mathbf{y}^{(1)}) &= \underbrace{L_J(\mathbf{y}^{(1)},\mathbf{y}^{(1)})}_{0}\bar{\mathbf{P}}(\mathbf{y}^{(1)}) + \underbrace{L_J(\mathbf{y}^{(2)},\mathbf{y}^{(1)})}_{1}\cdot\underbrace{\bar{\mathbf{P}}(\mathbf{y}^{(2)})}_{1/2} && \text{From (B.18)}\\ &= \frac{1}{2}.\\ \lim_{\epsilon\to 0} R_J(\mathbf{y}^{(1,2)}) &= \underbrace{L_J(\mathbf{y}^{(1)},\mathbf{y}^{(1,2)})}_{1/2}\cdot\underbrace{\bar{\mathbf{P}}(\mathbf{y}^{(1)})}_{1/2} + \underbrace{L_J(\mathbf{y}^{(2)},\mathbf{y}^{(1,2)})}_{1/2}\cdot\underbrace{\bar{\mathbf{P}}(\mathbf{y}^{(2)})}_{1/2} && \text{From (B.18)}\\ &= \frac{1}{2}.\end{aligned} \tag{B.19}$$

The optimal risk is $\frac{1}{2}$, when $\epsilon \to 0$. The risk of $\mathbf{h}^{\text{clr}}$ is given by

$$\begin{aligned}\lim_{\epsilon\to 0} R_J(\mathbf{h}^{\text{clr}}) &= \lim_{\epsilon\to 0} R_J(0_n) && \text{From Proposition 4.2}\\ &= \underbrace{L_J(\mathbf{y}^{(1)},0_n)}_{1}\cdot\bar{\mathbf{P}}(\mathbf{y}^{(1)}) + \underbrace{L_J(\mathbf{y}^{(2)},0_n)}_{1}\cdot\bar{\mathbf{P}}(\mathbf{y}^{(2)}) \\ &= 1.\end{aligned} \tag{B.20}$$

The regret is given by:

$$\begin{aligned}\lim_{\epsilon\to 0} r_J(\mathbf{h}^{\text{clr}}) &= \lim_{\epsilon\to 0} R_J(\mathbf{h}^{\text{clr}}) - \lim_{\epsilon\to 0} R_J(\mathbf{y}^*)\\ &= \frac{1}{2} && \text{From (B.19) and (B.20)}\end{aligned}$$

□

B.9 Proof of Theorem 4.8

Theorem 4.8. *The following expression holds for the regret with respect to F-measure loss*

$$\lim_{\epsilon\to 0} r_J(\mathbf{h}^{clr}) = \frac{2}{3}, \ \textit{for distribution } \bar{\mathbf{P}}.$$

This proof is similar to the proof of Theorem 4.7. Let A be a set of labellings of n labels defined as $A = \{0_n, \mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \mathbf{y}^{(1,2)}\}$. Like in Theorem 4.7, the risk on distribution $\bar{\mathbf{P}}$ can be expressed as (see Equation (B.18)):

$$\lim_{\epsilon\to 0} R_F(\hat{\mathbf{y}}) = L_F(\mathbf{y}^{(1)}, \hat{\mathbf{y}})\mathbf{P}(\mathbf{y}^{(1)}) + L_F(\mathbf{y}^{(2)}, \hat{\mathbf{y}})\mathbf{P}(\mathbf{y}^{(2)}).$$

An optimal solution for F-measure on $\bar{\mathbf{P}}$ is clearly either $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}$ or $\mathbf{y}^{(1,2)}$. This can be easily solved by checking all three values:

$$\begin{aligned}\lim_{\epsilon\to 0} R_F(\mathbf{y}^{(1)}) &= \underbrace{L_F(\mathbf{y}^{(1)}, \mathbf{y}^{(1)})}_{0} \cdot \bar{\mathbf{P}}(\mathbf{y}^{(1)}) + \underbrace{L_F(\mathbf{y}^{(2)}, \mathbf{y}^{(1)})}_{1} \cdot \underbrace{\bar{\mathbf{P}}(\mathbf{y}^{(2)})}_{1/2} \\ &= \frac{1}{2}. \\ R_F(\mathbf{y}^{(1,2)}) &= \underbrace{L_F(\mathbf{y}^{(1)}, \mathbf{y}^{(1,2)})}_{1/3} \cdot \underbrace{\bar{\mathbf{P}}(\mathbf{y}^{(1)})}_{1/2} + \underbrace{L_F(\mathbf{y}^{(2)}, \mathbf{y}^{(1,2)})}_{1/3} \cdot \underbrace{\bar{\mathbf{P}}(\mathbf{y}^{(2)})}_{1/2} \\ &= \frac{1}{3}.\end{aligned} \tag{B.21}$$

The risk of $\mathbf{h}^{\text{clr}}$ is given by

$$\begin{aligned}\lim_{\epsilon\to 0} R_F(\mathbf{h}^{\text{clr}}) &= \lim_{\epsilon\to 0} R_J(0_n) && \text{From Proposition 4.2} \\ &= \underbrace{L_F(\mathbf{y}^{(1)}, 0_n)}_{1} \cdot \bar{\mathbf{P}}(\mathbf{y}^{(1)}) + \underbrace{L_J(\mathbf{y}^{(2)}, 0_n)}_{1} \cdot \bar{\mathbf{P}}(\mathbf{y}^{(2)}) \\ &= 1.\end{aligned} \tag{B.22}$$

The regret is given by:

$$\begin{aligned}\lim_{\epsilon\to 0} r_F(\mathbf{h}^{\text{clr}}) &= \lim_{\epsilon\to 0} R_F(\mathbf{h}^{\text{clr}}) - \underbrace{\lim_{\epsilon\to 0} R_F(\mathbf{y}^{*})}_{1/3} \\ &= \frac{2}{3} && \text{From (B.21) and (B.22)}\end{aligned}$$

□

APPENDIX C – Appendices for Chapter 5

C.1 Proof of Equation 5.2

Theorem C.1. *For the simulation process described at Algorithm 3, the expected density of each label equals to* $\hat{\mu}$*:*

$$\mathbf{P}(Y_i = 1) = \hat{\mu}, \quad 1 \leq i \leq n$$

For any natural number m, let $\mathbf{Y}^{(m)}$ be the vector $(Y_1, ..., Y_m)$. The marginal distribution can be defined as:

$$\sum_{\mathbf{y} \in \{0,1\}^{(i-1)}} \mathbf{P}(Y_i = 1 | \mathbf{Y}^{(i-1)} = \mathbf{y}) \cdot \mathbf{P}(\mathbf{Y}^{(i-1)} = \mathbf{y}). \tag{C.1}$$

For any $1 \leq i \leq n$ let $B_{i\cdot}$ denote the corresponding node for the probability $\mathbf{P}(Y_i = 1 | \mathbf{Y}^{(i-1)} = \mathbf{y})$. The value $\mathbf{P}(Y_i = 1 | \mathbf{Y}^{(i-1)} = \mathbf{y})$ can be calculated as:

$$\begin{aligned} &\int_0^1 \mathbf{P}(Y_i = 1 | \mathbf{Y}^{(i-1)} = \mathbf{y}, B_{i\cdot} = x) \cdot \mathbf{P}(B_{i\cdot} = x) dx \\ = &\int_0^1 x \cdot \mathbf{P}(B_{i\cdot} = x) dx, \end{aligned}$$

which is by definition the expected value of $B_{i\cdot}$, therefore

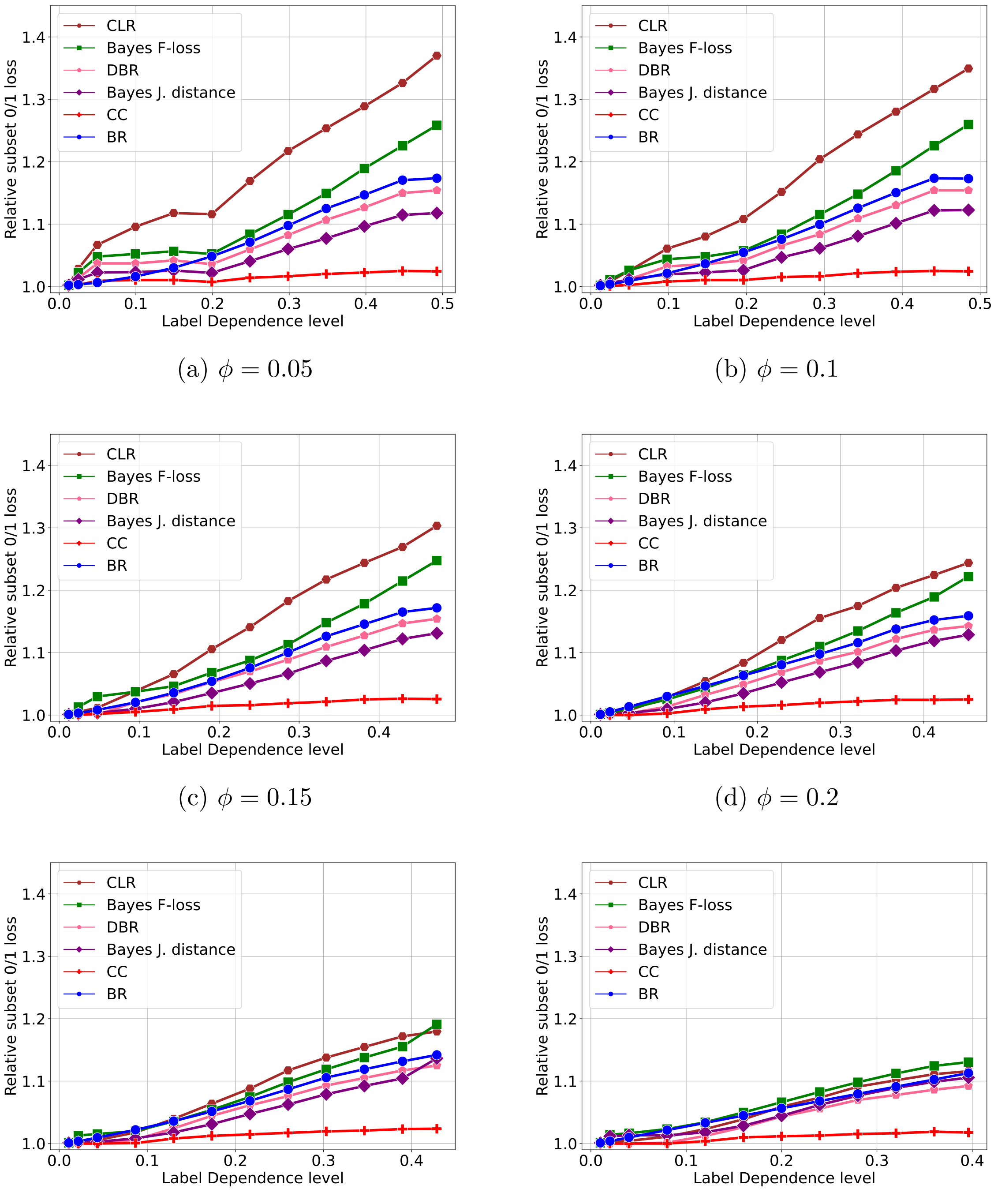
$$\begin{aligned} \mathbf{P}(Y_i = 1 | \mathbf{Y}^{(i-1)} = \mathbf{y}) &= \mathbb{E}\left[B_{ij}\right] \\ &= \mathbb{E}_{\mu_i}\left[\mathbb{E}[B_{ij} | \mu_i]\right] \\ &= \mathbb{E}_{\mu_i}\left[\mu_i\right] \\ &= \hat{\mu}. \end{aligned} \tag{C.2}$$

From (C.1) and (C.2):

$$\begin{aligned} \mathbf{P}(Y_i = 1) &= \sum_{\mathbf{y} \in \{0,1\}^{(i-1)}} \hat{\mu} \cdot \mathbf{P}(\mathbf{Y}^{(i-1)} = \mathbf{y}) \\ &= \hat{\mu} \cdot \sum_{\mathbf{y} \in \{0,1\}^{(i-1)}} \mathbf{P}(\mathbf{Y}^{(i-1)} = \mathbf{y}) \\ &= \hat{\mu}. \end{aligned}$$

□

C.2 Additional Graphs for subset 0/1 loss



(a) $\phi = 0.05$

(b) $\phi = 0.1$

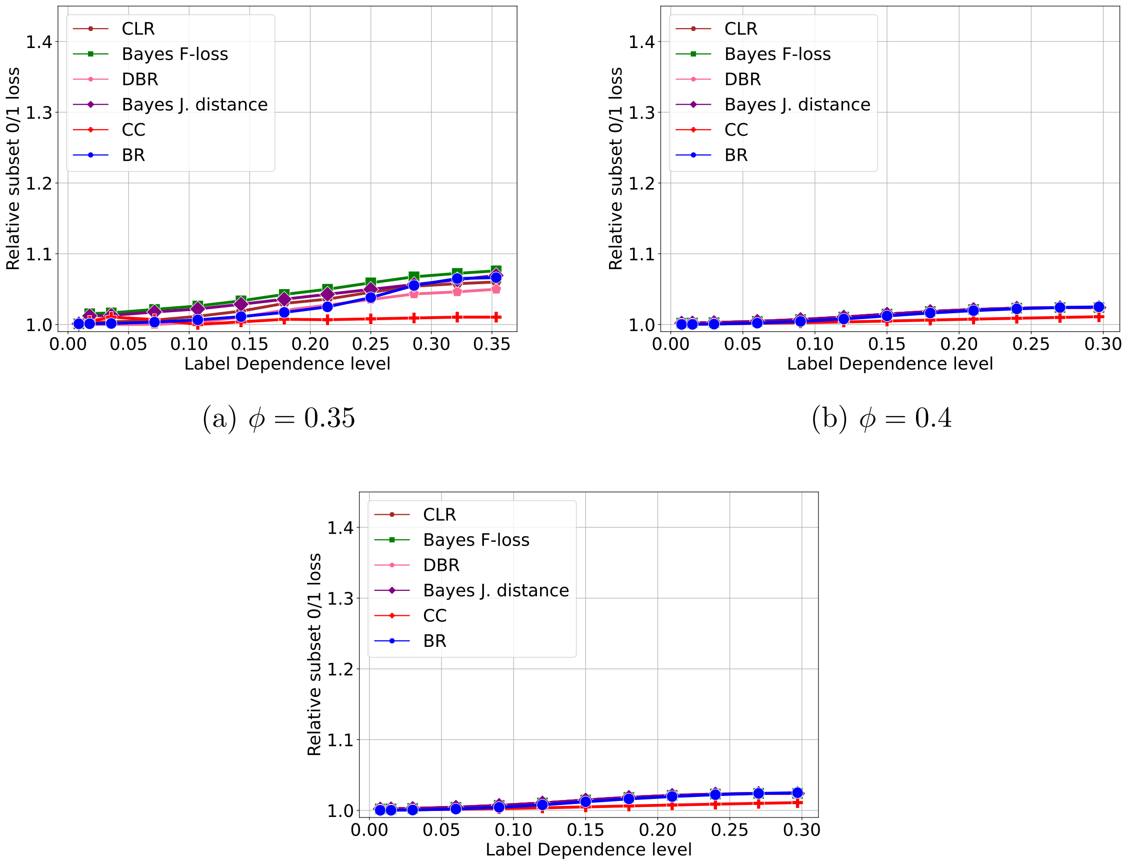
(c) $\phi = 0.15$

(d) $\phi = 0.2$

(e) $\phi = 0.25$

(f) $\phi = 0.3$

Figure 8 – Graphs of estimated expected subset 0/1 loss for all methods divided by the best one (PCC). The value of γ is presented on the horizontal axis.



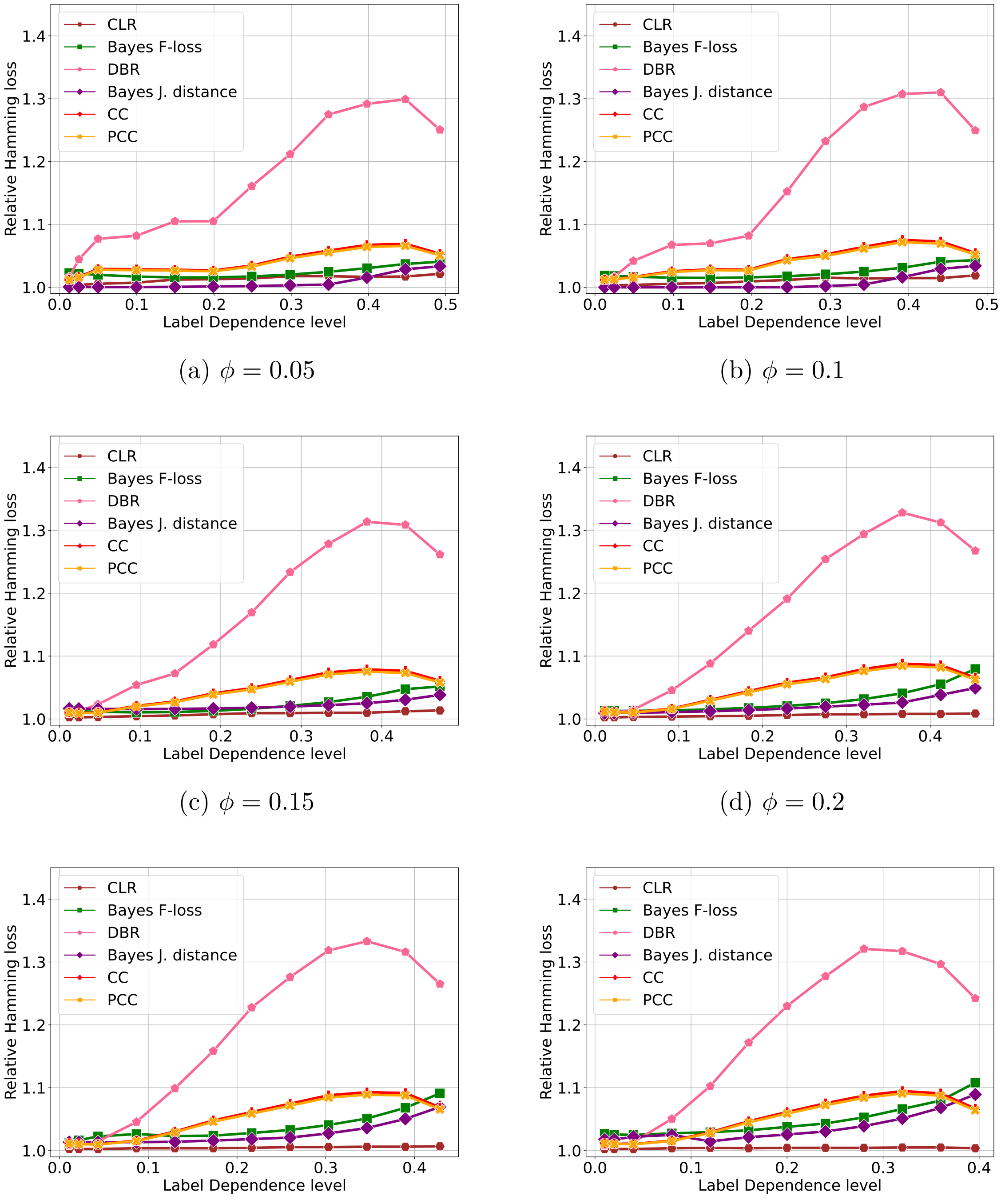
(a) $\phi = 0.35$

(b) $\phi = 0.4$

(c) $\phi = 0.45$

Figure 9 – Graphs of estimated expected subset 0/1 loss for all methods divided by the best one (PCC). The value of γ is presented on the horizontal axis.

C.3 Additional Graphs for Hamming loss



(a) $\phi = 0.05$

(b) $\phi = 0.1$

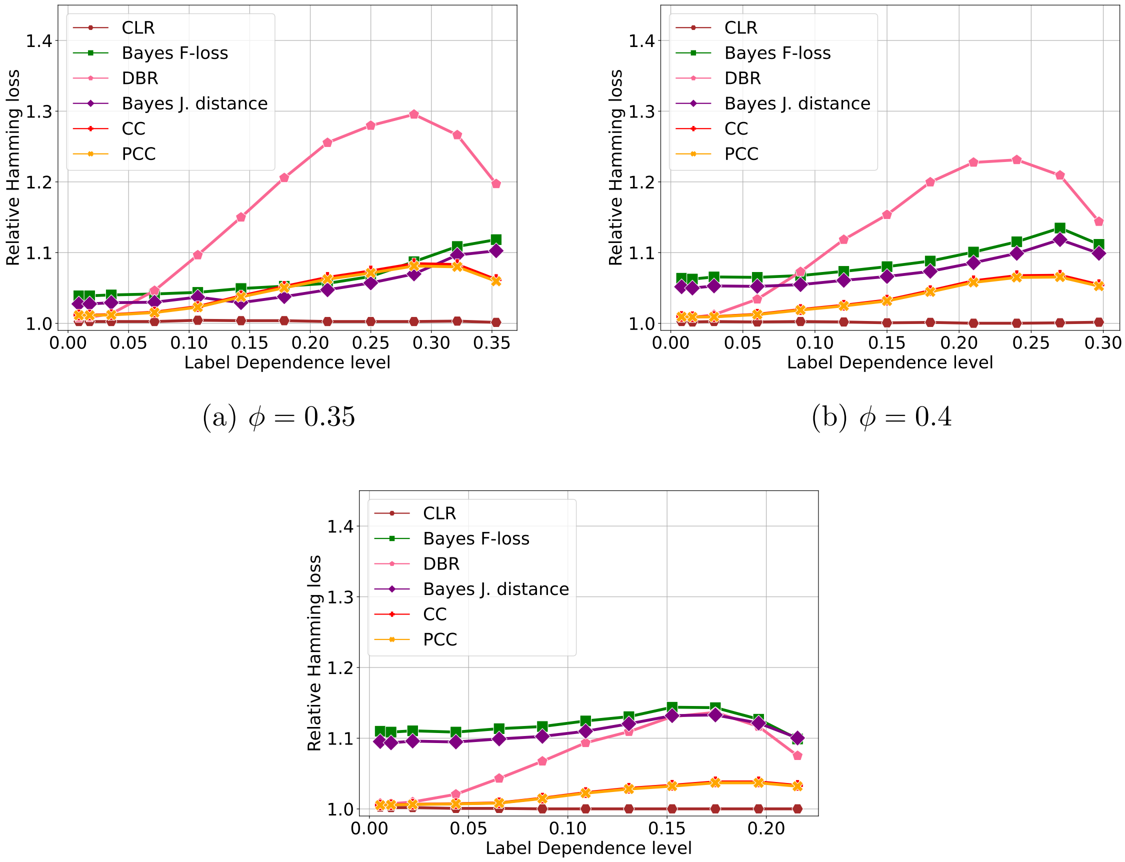
(c) $\phi = 0.15$

(d) $\phi = 0.2$

(e) $\phi = 0.25$

(f) $\phi = 0.3$

Figure 10 – Graphs of estimated expected Hamming loss for all methods divided by the best one (BR). The value of γ is presented on the horizontal axis.



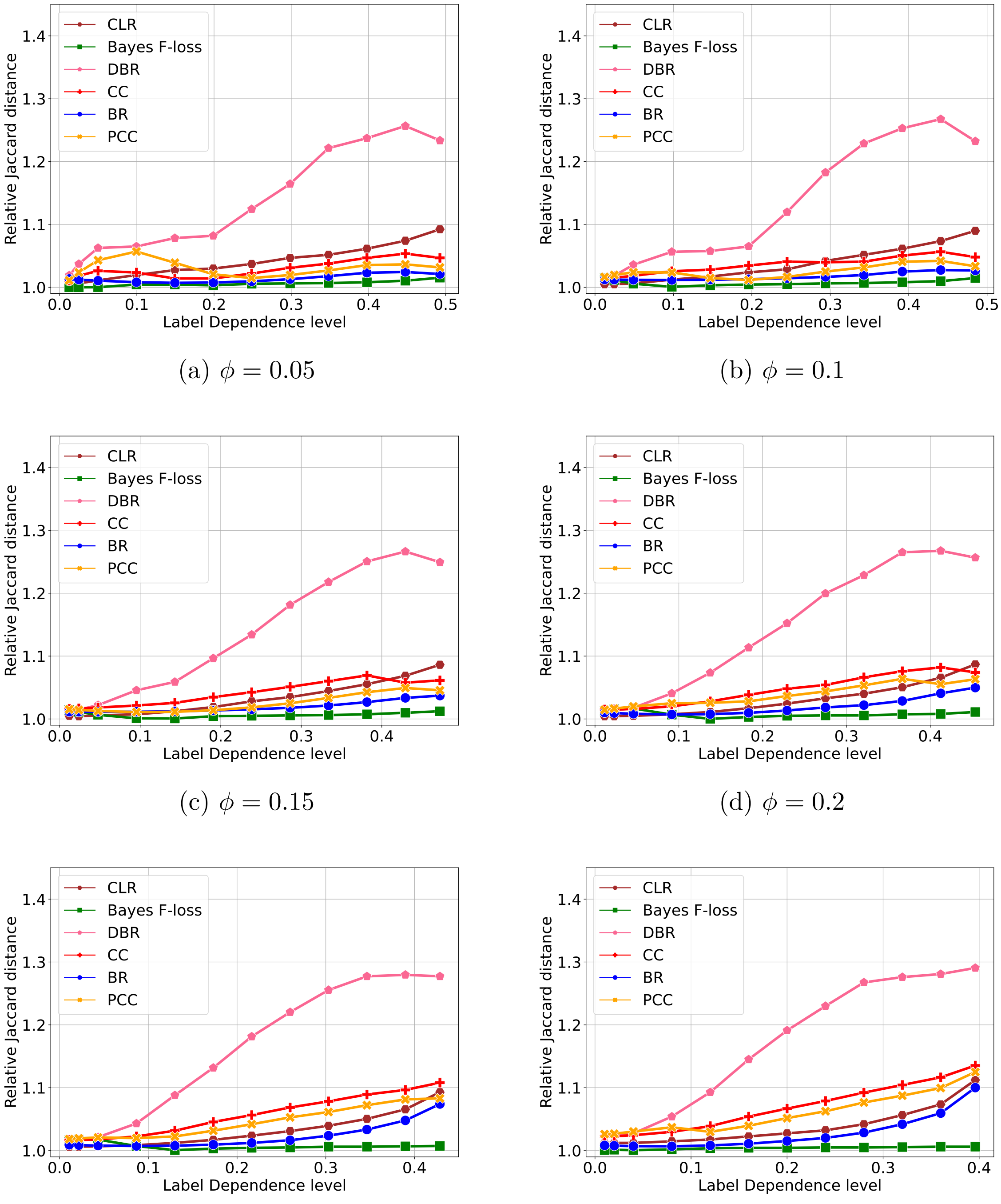
(a) $\phi = 0.35$

(b) $\phi = 0.4$

(c) $\phi = 0.45$

Figure 11 – Graphs of estimated expected Hamming loss for all methods divided by the best one (BR). The value of γ is presented on the horizontal axis.

C.4 Additional Graphs for Jaccard distance



(a) $\phi = 0.05$

(b) $\phi = 0.1$

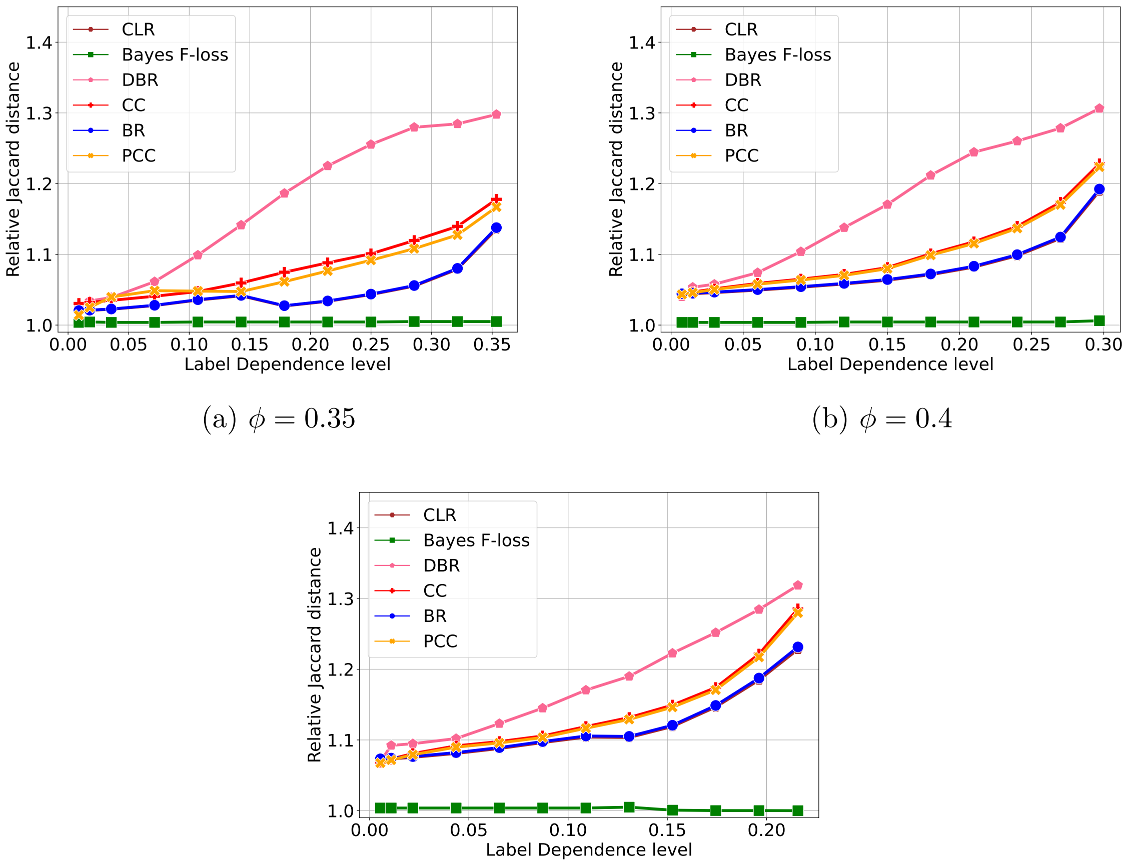
(c) $\phi = 0.15$

(d) $\phi = 0.2$

(e) $\phi = 0.25$

(f) $\phi = 0.3$

Figure 12 – Graphs of estimated expected Jaccard distance for all methods divided by the best one (Bayes J. distance). The value of γ is presented on the horizontal axis.



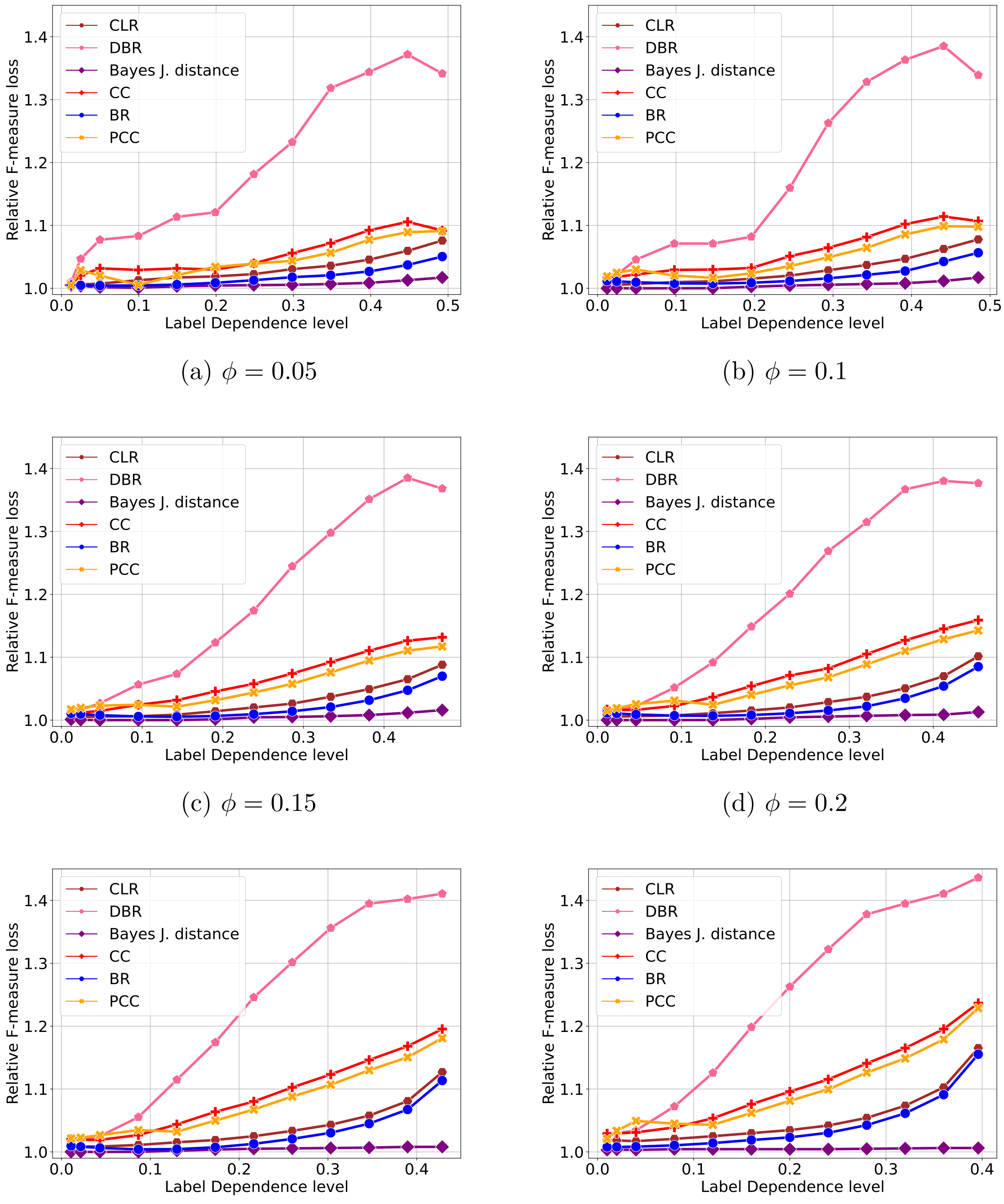
(a) $\phi = 0.35$

(b) $\phi = 0.4$

(c) $\phi = 0.45$

Figure 13 – Graphs of estimated expected Jaccard distance for all methods divided by the best one (Bayes J. distance). The value of γ is presented on the horizontal axis.

C.5 Additional Graphs for F-measure



(a) $\phi = 0.05$

(b) $\phi = 0.1$

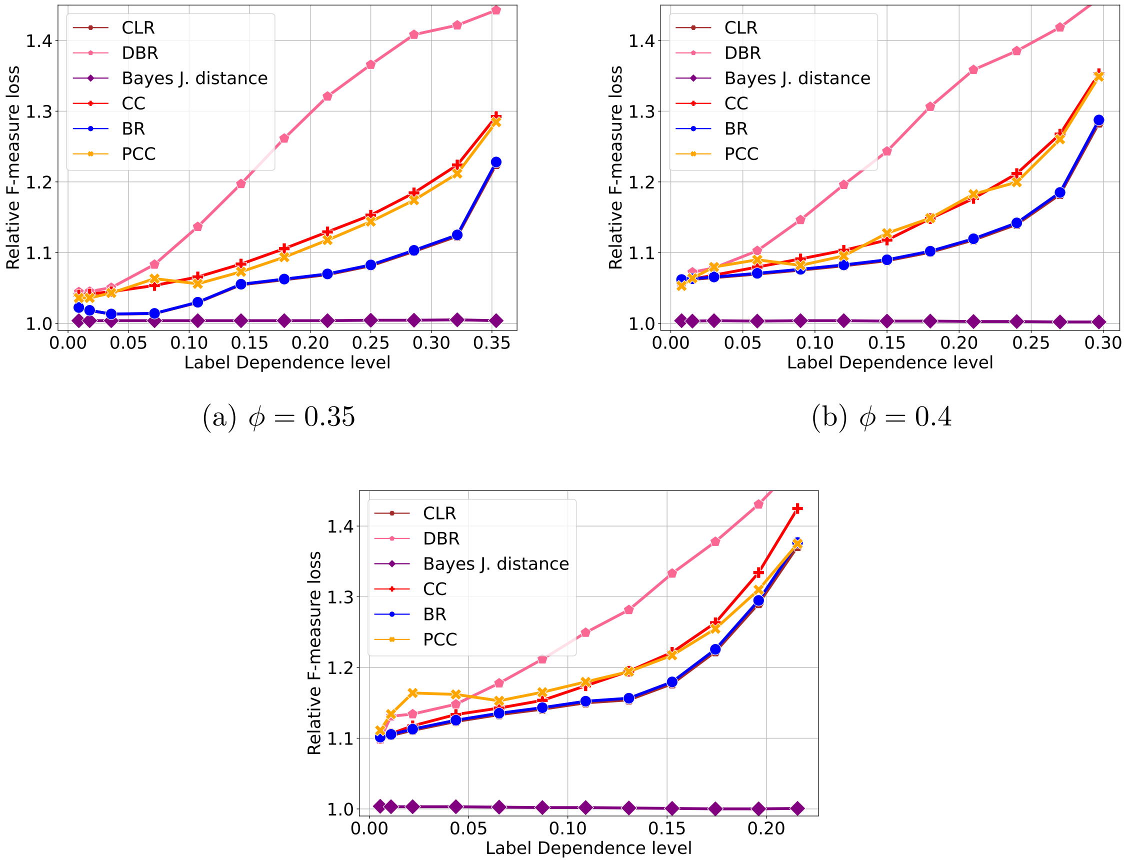
(c) $\phi = 0.15$

(d) $\phi = 0.2$

(e) $\phi = 0.25$

(f) $\phi = 0.3$

Figure 14 – Graphs of estimated expected F-measure for all methods divided by the best one (Bayes F-loss). The value of γ is presented on the horizontal axis.



(a) $\phi = 0.35$

(b) $\phi = 0.4$

(c) $\phi = 0.45$

Figure 15 – Graphs of estimated expected F-measure for all methods divided by the best one (Bayes F-loss). The value of γ is presented on the horizontal axis.

References

ARTEM, P.; ALOISE, D.; MLADENOVIĆ, N. NP-hardness of balanced minimum sum-of-squares clustering. *Pattern Recognition Letters*, p. 1–2, jun. 2017. Cited on page 32.

BLANCHARD, T.; LOMBROZO, T.; NICHOLS, S. Bayesian occam's razor is a razor of the people. *Cognitive science*, Wiley Online Library, v. 42, n. 4, p. 1345–1359, 2018. Cited on page 51.

CARVALHO, A. C.; FREITAS, A. A. A tutorial on multi-label classification techniques. In: *Foundations of Computational Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, (Studies in Computational Intelligence, v. 5). p. 177–195. Cited on page 13.

CHEN, H.; KARGER, D. R. Less is more: Probabilistic models for retrieving fewer relevant documents. In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: Association for Computing Machinery, 2006. (SIGIR '06), p. 429–436. Cited on pages 27, 31, and 32.

CHENG, W.; HÜLLERMEIER, E. Combining instance-based learning and logistic regression for multilabel classification. In: BUNTINE, W. et al. (Ed.). *Machine Learning and Knowledge Discovery in Databases*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 6–6. Cited on page 59.

CHENG, W.; HÜLLERMEIER, E.; DEMBCZYŃSKI, K. J. Bayes optimal multilabel classification via probabilistic classifier chains. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. [S.l.: s.n.], 2010. p. 279–286. Cited on page 28.

CHIERICHETTI, F. et al. Finding the jaccard median. In: *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*. USA: Society for Industrial and Applied Mathematics, 2010. (SODA '10), p. 293–311. Cited on pages 28 and 31.

COOPER, W. S. Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American Documentation*, v. 19, n. 1, p. 30–41, 1968. Cited on page 27.

DECUBBER, S. et al. Deep f-measure maximization in multi-label classification: A comparative study: Recognizing outstanding ph.d. research. In: _____. [S.l.: s.n.], 2019. p. 290–305. ISBN 978-981-13-6048-0. Cited on page 59.

DEMBCZYŃSKI, K.; CHENG, W.; HÜLLERMEIER, E. Bayes optimal multilabel classification via probabilistic classifier chains. In: FÜRNKRANZ, J.; JOACHIMS, T. (Ed.). *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. Haifa, Israel: Omnipress, 2010. p. 279–286. Cited on pages 14, 19, 20, 21, 31, 54, and 65.

DEMBCZYŃSKI, K. et al. Optimizing the f-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In: DASGUPTA, S.; MCALLESTER,

D. (Ed.). *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. [S.l.]: JMLR Workshop and Conference Proceedings, 2013. p. 1130–1138. Cited on pages 14, 22, 31, 32, 65, and 67.

DEMBCZYŃSKI, K.; KOTŁOWSKI, W.; HÜLLERMEIER, E. Consistent multilabel ranking through univariate loss minimization. In: *Proceedings of the 29th International Conference on International Conference on Machine Learning*. Madison, WI, USA: Omnipress, 2012. (ICML'12), p. 1347–1354. Cited on page 28.

DEMBCZYŃSKI, K. et al. On label dependence and loss minimization in multi-label classification. *Machine Learning*, Springer, v. 88, p. 5–45, 2012. Cited on pages 14, 28, 32, 49, 51, 53, 54, 55, 56, and 69.

DOPPA, J. R. et al. Hc-search for multi-label prediction: An empirical study. In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. [S.l.]: AAAI Press, 2014. (AAAI'14), p. 1795–1801. Cited on page 65.

FÜRNKRANZ, J. et al. Multilabel classification via calibrated label ranking. *Machine Learning*, v. 73, n. 2, p. 133–153, Nov 2008. Cited on pages 20, 24, and 47.

GENG, X. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 28, n. 7, p. 1734–1748, 2016. Cited on page 19.

HUANG, M. et al. Supervised representation learning for multi-label classification. *Machine Learning*, v. 108, n. 5, p. 747–763, 2019. Cited on pages 45 and 47.

Hüllermeier, E.; Furnkranz, J. Ranking by pairwise comparison a note on risk minimization. In: *2004 IEEE International Conference on Fuzzy Systems*. [S.l.]: IEEE, 2004. v. 1, p. 97–102. Cited on pages 27, 29, and 52.

JASINSKA-KOBUS, K. et al. Probabilistic label trees for extreme multi-label classification. *arXiv preprint*, 2020. Cited on page 31.

JEFFERYS, W. H.; BERGER, J. O. Ockham's razor and bayesian analysis. *American Scientist*, JSTOR, v. 80, n. 1, p. 64–72, 1992. Cited on page 51.

JIA, X. et al. Label distribution learning by exploiting label correlations. In: *Thirty-Second AAAI Conference on Artificial Intelligence*. North America: AAAI, 2018. Cited on page 19.

JIANG, C.-R. et al. Optimal ranking in multi-label classification using local precision rates. *Statistica Sinica*, v. 24, n. 4, p. 1547–1570, 2014. Cited on pages 53, 56, and 57.

KARP, R. Reducibility among combinatorial problems. In: *Complexity of Computer Computations*. Boston, MA: Plenum Press, 1972. p. 85–103. Cited on page 36.

MADJAROV, G. et al. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, v. 45, n. 9, p. 3084–3104, 2012. Cited on page 31.

MCCALLUM, A. K. Multi-label text classification with a mixture model trained by em. In: CITESEER. *AAAI 99 workshop on text learning*. [S.l.], 1999. Cited on page 13.

MELLO, L. H. S. et al. Metric learning for electrical submersible pump fault diagnosis. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2020. p. 1–8. Cited on page 15.

MELLO, L. H. S. et al. NP-hardness of minimum expected coverage. *Pattern Recognition Letters*, v. 117, p. 45 – 51, 2019. Cited on page 14.

MICHALSKI, R. S.; BRATKO, I.; BRATKO, A. *Machine Learning and Data Mining; Methods and Applications*. USA: John Wiley & Sons, Inc., 1998. Cited on page 13.

Min-Ling Zhang; Zhi-Hua Zhou. A k-nearest neighbor based algorithm for multi-label classification. In: *2005 IEEE International Conference on Granular Computing*. USA: IEEE, 2005. v. 2, p. 718–721. Cited on page 59.

MONTAÑES, E. et al. Dependent binary relevance models for multi-label classification. *Pattern Recognition*, v. 47, n. 3, p. 1494 – 1508, 2014. Cited on pages 20 and 68.

NOH, H. G.; SONG, M. S.; PARK, S. H. An unbiased method for constructing multilabel classification trees. *Computational Statistics & Data Analysis*, v. 47, n. 1, p. 149–164, ago. 2004. Cited on pages 14, 55, and 56.

PEREIRA, R. B. et al. Correlation analysis of performance measures for multi-label classification. *Information Processing and Management*, v. 54, n. 3, p. 359 – 369, 2018. Cited on pages 56, 67, and 69.

READ, J. et al. Classifier chains for multi-label classification. In: *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*. Berlin, Heidelberg: Springer-Verlag, 2009. (ECML PKDD '09), p. 254–269. Cited on page 20.

SCHAPIRE, R. E.; SINGER, Y. Boostexter: A boosting-based system for text categorization. *Machine Learning*, v. 39, n. 2, p. 135–168, 2000. Cited on pages 27, 31, and 32.

SCHÖNING, U.; PRUIM, R. J. *Gems of theoretical computer science*. German: Springer Science & Business Media, 2012. Cited on page 52.

SENGE, R.; COZ, J. J. del; HÜLLERMEIER, E. Rectifying classifier chains for multi-label classification. In: HENRICH, A.; SPERKER, H. (Ed.). *LWA 2013. Lernen, Wissen & Adaptivität, Workshop Proceedings Bamberg*. [S.l.]: Universitätsbibliothek Bamberg, 2013. p. 151–158. Cited on page 54.

SETTLES, B. Active learning literature survey. University of Wisconsin-Madison Department of Computer Sciences, 2009. Cited on page 31.

SOROWER, M. S. *A literature survey on algorithms for multi-label learning*. [S.l.], 2010. Cited on page 31.

SULAIMAN, M. et al. Application of beta distribution model to malaysian sunshine data. *Renewable Energy*, v. 18, n. 4, p. 573 – 579, 1999. Cited on pages 63 and 70.

SUN, L.; GE, H.; KANG, W. Non-negative matrix factorization based modeling and training algorithm for multi-label learning. *Frontiers of Computer Science*, v. 13, n. 6, p. 1243–1254, 2019. Cited on page 47.

SUN, L.; KUDO, M. Optimization of classifier chains via conditional likelihood maximization. *Pattern Recognition*, v. 74, p. 503 – 517, 2018. Cited on page 19.

TAHIR, M.; KITTLER, J.; BOURIDANE, A. Multi-label classification using stacked spectral kernel discriminant analysis. *Neurocomputing*, v. 171, p. 127–137, 2016. Cited on pages 45 and 47.

TEUGELS, J. L. Some representations of the multivariate bernoulli and binomial distributions. *Journal of multivariate analysis*, Elsevier, v. 32, n. 2, p. 256–268, 1990. Cited on pages 39, 40, and 49.

TOMÁS, J. T. et al. A framework to generate synthetic multi-label datasets. *Electronic Notes in Theoretical Computer Science*, v. 302, p. 155 – 176, 2014. Latin American Computing Conference (CLEI). Cited on pages 14, 54, and 55.

TROHIDIS, K. et al. Multi-label classification of music by emotion. *EURASIP Journal on Audio, Speech, and Music Processing*, v. 2011, n. 1, p. 4, Sep 2011. Cited on pages 45 and 47.

TSOUMAKAS, G.; KATAKIS, I. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, p. 1–13, 2007. Cited on pages 13 and 20.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Mining multi-label data. In: *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer US, 2010. p. 667–685. Cited on pages 13, 17, 19, 20, and 22.

WAEGEMAN, W. et al. On the bayes-optimality of f-measure maximizers. *Journal of Machine Learning Research*, v. 15, p. 3513–3568, 2014. Cited on pages 14, 53, 54, and 57.

WANG, J.; GENG, X. Theoretical analysis of label distribution learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 33, p. 5256–5263, 07 2019. Cited on page 19.

WANG, S. et al. Enhancing multi-label classification by modeling dependencies among labels. *Pattern Recognition*, Elsevier, v. 47, n. 10, p. 3405–3413, 2014. Cited on pages 45 and 47.

YE, N. et al. Optimizing f-measure: A tale of two approaches. In: *International Conference on Machine Learning*. Madison, WI, USA: Omnipress, 2012. Cited on page 56.

YOUNES, Z. et al. A dependent multilabel classification method derived from the k-nearest neighbor rule. *EURASIP Journal on Advances in Signal Processing*, Springer, p. 1–14, 2011. Cited on page 59.

ZHAI, C.; COHEN, W. W.; LAFFERTY, J. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. *SIGIR Forum*, Association for Computing Machinery, New York, NY, USA, v. 49, n. 1, p. 2–9, jun. 2003. Cited on page 31.

ZHANG, M.; ZHOU, Z. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, v. 26, p. 1819–1837, 2013. Cited on pages 19, 59, and 65.

ZHANG, M.-L.; PEÑA, J. M.; ROBLES, V. Feature selection for multi-label naive bayes classification. *Information Sciences*, v. 179, n. 19, p. 3218–3229, 2009. Cited on page 55.

ZHANG, M.-L.; ZHANG, K. Multi-label learning by exploiting label dependency. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2010. (KDD '10), p. 999–1008. Cited on page 31.

ZHANG, Y.; SCHNEIDER, J. A composite likelihood view for multi-label classification. In: LAWRENCE, N. D.; GIROLAMI, M. (Ed.). *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*. La Palma, Canary Islands: PMLR, 2012. (Proceedings of Machine Learning Research, v. 22), p. 1407–1415. Cited on pages 45 and 47.

ZHANG, Y.; SCHNEIDER, J. Maximum margin output coding. In: *Proceedings of the 29th International Coference on International Conference on Machine Learning*. Madison, WI, USA: Omnipress, 2012. (ICML'12), p. 379–386. Cited on pages 45 and 47.