Journal of Artificial Intelligence Research 72 (2021) 613-665 Submitted 01/2021; published 11/2021

Multilabel Classification with Partial Abstention: Bayes-Optimal Prediction under Label Independence

Vu-Linh Nguyen V.L.NGUYEN@TUE.NL
Department of Mathematics and Computer Science
Eindhoven University of Technology, The Netherlands

Eyke Hüllermeier EYKE@LMU.DE
Department of Computer Science
University of Munich (LMU), Germany

Abstract

In contrast to conventional (single-label) classification, the setting of *multilabel classification* (MLC) allows an instance to belong to several classes simultaneously. Thus, instead of selecting a single class label, predictions take the form of a subset of all labels. In this paper, we study an extension of the setting of MLC, in which the learner is allowed to partially abstain from a prediction, that is, to deliver predictions on some but not necessarily all class labels. This option is useful in cases of uncertainty, where the learner does not feel confident enough on the entire label set. Adopting a decision-theoretic perspective, we propose a formal framework of MLC with partial abstention, which builds on two main building blocks: First, the extension of underlying MLC loss functions so as to accommodate abstention in a proper way, and second the problem of optimal prediction, that is, finding the Bayes-optimal prediction minimizing this generalized loss in expectation. It is well known that different (generalized) loss functions may have different risk-minimizing predictions, and finding the Bayes predictor typically comes down to solving a computationally complexity optimization problem. In the most general case, given a prediction of the (conditional) joint distribution of possible labelings, the minimizer of the expected loss needs to be found over a number of candidates which is exponential in the number of class labels. We elaborate on properties of risk minimizers for several commonly used (generalized) MLC loss functions, show them to have a specific structure, and leverage this structure to devise efficient methods for computing Bayes predictors. Experimentally, we show MLC with partial abstention to be effective in the sense of reducing loss when being allowed to abstain.

1. Introduction

In statistics and machine learning, classification with abstention, also known as classification with a reject option, is an extension of the standard setting of classification, in which the learner is allowed to refuse a prediction for a given query instance; research on this setting dates back to early work by Chow (1970) and Hellman (1970) and remains to be an important topic till today, most notably for binary classification (Bartlett and Wegkamp, 2008; Cortes et al., 2016; Franc and Prusa, 2019; Grandvalet et al., 2008). For the learner, the main reason to abstain is a lack of certainty about the corresponding outcome — refusing or at least deferring a decision might then be better than taking a high risk of a wrong decision.

Nowadays, there are many machine learning problems in which complex, structured predictions are sought (instead of scalar values, like in classification and regression). For such problems, the idea of abstaining from a prediction can be generalized toward *partial abstention*: Instead of predicting the entire structure, the learner predicts only parts of it, namely those for which it is certain enough. This idea has already been realized, for example, for the problem of *label ranking*, where predictions appear in the form of total orders of a set of class labels (Cheng et al., 2010, 2012).

Another important example is *multilabel classification* (MLC), in which an outcome associated with an instance is a labeling in the form of a subset of an underlying reference set of class labels; that is, the output space is the power set of that reference set (Tsoumakas et al., 2009; Zhang and Zhou, 2014). MLC problems naturally occur in a variety of fields, such as text categorization (Hayes and Weinstein, 1990; Lewis, 1992), music categorization (Trohdis, 2008), semantic scene classification (Boutell et al., 2004), protein function classification Elisseeff and Weston (2001), or functional genomics and text categorization (Zhang and Zhou, 2006). In this paper, we study an extension of the setting of MLC, in which the learner is allowed to partially abstain from a prediction, that is, to deliver predictions on some but not necessarily all class labels (or, more generally, to refuse committing to a single complete prediction). Although MLC has been studied extensively in the machine learning literature in the recent past, there is surprisingly little work on MLC with abstention so far — a notable exception is the work of Pillai et al. (2013), to which we shall return in Section 8.

Prediction with abstention is typically realized as a two-stage approach. First, the learner delivers a prediction that provides information about its uncertainty. Then, taking this uncertainty into account, a decision is made about whether or not to predict, or on which parts. In binary classification, for example, a typical approach is to produce probabilistic predictions and to abstain whenever the probability is close to $1/2$, which is considered as the case of maximal uncertainty. We adopt a similar approach, in which we rely on probabilistic MLC, i.e., probabilistic predictions of labelings. More specifically, we follow the decision-theoretic approach, in which the problem is formalized as finding the Bayes-optimal prediction (BOP), that is, the prediction that minimizes the expected loss, where the expectation is taken with respect to the given probability distribution on the labeling space (Dembczyński et al., 2012; Waegeman et al., 2014; Ye et al., 2012).

Of course, the BOP does not only depend on the probability distribution on labelings, but also on the underlying loss function used to assess multilabel predictions. Moreover, it is well known that different loss functions may call for different BOPs, and the corresponding optimization problem of finding the BOP can be computationally demanding (Dembczyński et al., 2012). In the most general case, it requires the expectation to be computed over 2^K candidate predictions, where K is the number of class labels. Fortunately, by exploiting structural properties of the BOP, this candidate set can be significantly reduced, at least under certain assumptions. In this paper, we exploit the property of (conditional) *label independence* (Dembczyński et al., 2012), which, roughly speaking, stipulates that the probability of label co-occurrence is the product of the individual probabilities of occurrence. For many common MLC loss functions, we show that, under this assumption, the number of candidate predictions can be reduced from exponential to quadratic in K. Based on theoretical results of this kind, we devise efficient algorithms for Bayes-optimal prediction in the setting of MLC with partial abstention.

The organization of the paper is as follows. In the next section, we briefly recall the setting of multilabel classification. The generalization toward MLC with partial abstention is then introduced and formalized in Section 3. Instantiations of this setting for various MLC loss functions are studied in Sections 4–7, and related work is discussed in Section 8. Finally, experimental results are presented in Section 9, prior to concluding the paper in Section 10.

This paper is an extension of an earlier conference version (Nguyen and Hüllermeier, 2020), in which the setting of MLC with partial abstention has originally been introduced and studied for selected loss functions. The current version is more comprehensive in a number of ways, especially regarding the class of loss functions covered and the experimental evaluation. All formal results in this paper (lemmas, propositions, remarks, corollaries) are stated without proofs, which are deferred to the appendix.

symbol/acronym	meaning
$\mathcal{X}$, $\boldsymbol{x}$	instance space, instance
$\mathcal{L}$, λ_k	label space, label
$\Lambda(\boldsymbol{x})$	subset of labels associated with $\boldsymbol{x}$
$\mathcal{Y}$, $\boldsymbol{y}$	labeling space, labeling vector
$\mathcal{Y}^*$	space of partial predictions
K	number of labels
$[\![\cdot]\!]$	indicator function
$[n]$	set $\{1, \dots, n\}$ of natural numbers
$\boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x})$	probability of labeling $\boldsymbol{y}$ given $\boldsymbol{x}$
$p_k = \boldsymbol{p}_k(1 \mid \boldsymbol{x})$	marginal probability of relevance for label λ_k
$\mathcal{D}$	training data
$\hat{\boldsymbol{y}} = \boldsymbol{h}(\boldsymbol{x})$	labeling predicted by MLC classifier $\boldsymbol{h}$ for $\boldsymbol{x}$
$\mathcal{H}$	hypothesis space
$A(\hat{\boldsymbol{y}})$, $D(\hat{\boldsymbol{y}})$	abstention set, prediction set
ℓ, L	MLC loss function, extended loss function
f	performance metric (defined on confusion matrix)
g	function to penalize abstention
$\mathbf{E}$	expected value
π	permutation representing a label ranking $\lambda_{\pi(1)} \succ \dots \succ \lambda_{\pi(K)}$
$\pi(i)$	index of the label on position i of the ranking π
$\pi^{-1}(j)$	position of the label λ_j in the permutation π
BOP	Bayes-optimal prediction
DTA	decision-theoretic approach
CMDP	confusion matrix-derived performance measures
CLI	conditional label independence
BR	binary relevance learning

Table 1: Notation and acronyms

2. Multilabel Classification

In this section, we provide a formal description of the MLC problem and introduce the notation used throughout the paper.

2.1 General Setting

Let $\mathcal{X}$ denote an instance space, and let $\mathcal{L} = \{\lambda_1, \dots, \lambda_K\}$ be a finite set of class labels. We assume that an instance $\boldsymbol{x} \in \mathcal{X}$ is (probabilistically) associated with a subset of labels $\Lambda = \Lambda(\boldsymbol{x}) \in 2^{\mathcal{L}}$. The subset $\Lambda(\boldsymbol{x})$ is often called the set of relevant labels, while the complement $\mathcal{L} \setminus \Lambda(\boldsymbol{x})$ is considered as irrelevant for $\boldsymbol{x}$; alternatively, we say that the labels in $\Lambda(\boldsymbol{x})$ occurred (for $\boldsymbol{x}$), whereas those in the complement did not. We identify a set Λ of relevant labels with a binary vector $\boldsymbol{y} = (y_1, \dots, y_K)$, where $y_k = [\![\lambda_k \in \Lambda]\!]$[1]. By $\mathcal{Y} = \{0, 1\}^K$ we denote the labeling space, i.e., set of possible labelings.

We assume observations to be realizations of independently and identically distributed (i.i.d.) random variables generated according to a probability distribution $\boldsymbol{p}$ on $\mathcal{X} \times \mathcal{Y}$, i.e., an observation $\boldsymbol{y} = (y_1, \dots, y_K)$ is the realization of a corresponding random vector $\mathbf{Y} = (Y_1, \dots, Y_K)$. We denote by $\boldsymbol{p}(\mathbf{Y} \mid \boldsymbol{x})$ the conditional distribution of $\mathbf{Y}$ given $\mathbf{X} = \boldsymbol{x}$, and by $\boldsymbol{p}_k(Y_k \mid \boldsymbol{x})$ the corresponding marginal distribution of Y_k:

$$\boldsymbol{p}_k(b \mid \boldsymbol{x}) := \sum_{\boldsymbol{y} \in \mathcal{Y} : y_k = b} \boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x}) . \tag{1}$$

Moreover, we denote by $p_k := \boldsymbol{p}_k(1 \mid \boldsymbol{x})$ the probability of relevance of the label λ_k.

Given training data in the form of a finite set of observations

$$\mathcal{D} = \big\{ (\boldsymbol{x}_n, \boldsymbol{y}_n) \big\}_{n=1}^{N} \subseteq \mathcal{X} \times \mathcal{Y} , \tag{2}$$

drawn independently from $\boldsymbol{p}(\mathbf{X}, \mathbf{Y})$, the goal in MLC is to learn a predictive model in the form of a multilabel classifier $\boldsymbol{h}$, which is a mapping $\mathcal{X} \longrightarrow \mathcal{Y}$ that assigns a (predicted) label subset to each instance $\boldsymbol{x} \in \mathcal{X}$. Thus, the output of a classifier $\boldsymbol{h}$ is a vector

$$\boldsymbol{h}(\boldsymbol{x}) = \big(h_1(\boldsymbol{x}), \dots, h_K(\boldsymbol{x}) \big) \in \{0, 1\}^K . \tag{3}$$

Predictions of this kind will also be denoted as $\hat{\boldsymbol{y}} = \boldsymbol{h}(\boldsymbol{x}) = (\hat{y}_1, \dots, \hat{y}_K)$.

To evaluate the performance of a multilabel classifier $\boldsymbol{h}$, an MLC loss function

$$\ell : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}_+ \tag{4}$$

is needed, which compares a prediction $\boldsymbol{h}(\boldsymbol{x})$ with a ground-truth labeling $\boldsymbol{y}$. The prediction accuracy of $\boldsymbol{h}$ is measured in terms of its risk, that is, its expected loss

$$R(\boldsymbol{h}) := \mathbf{E}\big[\ell(\mathbf{Y}, \boldsymbol{h}(\mathbf{X})) \big] = \int \ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x}))\, d\, \mathbf{P}(\boldsymbol{x}, \boldsymbol{y}) ,$$

where $\mathbf{P}$ is the joint probability measure on $\mathcal{X} \times \mathcal{Y}$ characterizing the underlying data-generating process. Therefore, the Bayes-optimal (risk-minimizing) classifier is given by

$$\boldsymbol{h}^* := \operatorname*{argmin}_{\boldsymbol{h} \in \mathcal{H}} R(\boldsymbol{h}) , \tag{5}$$

where $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is the hypothesis space, i.e., the class of functions from which a predictor can be chosen. A common approach to learning the Bayes-optimal classifier (5) is based on the principle of empirical loss minimization (sometimes also called empirical utility maximization, if predictions are assessed in terms of a utility instead of a loss function), which essentially comes down to finding the hypothesis with the smallest empirical risk, i.e., average loss on the training data:

$$\hat{\boldsymbol{h}} := \operatorname*{argmin}_{\boldsymbol{h} \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \ell(\boldsymbol{y}_n, \boldsymbol{h}(\boldsymbol{x}_n)) .$$

[1] $[\![\cdot]\!]$ is the indicator function, i.e., $[\![A]\!] = 1$ if the predicate A is true and $= 0$ otherwise.

To prevent the learner from overfitting the training data, the empirical risk is often augmented by a regularization term.

As an alternative, a decision-theoretic approach (DTA) can be used, in which a hypothesis in the form of a mapping from instances $\boldsymbol{x} \in \mathcal{X}$ to predictions $\boldsymbol{y} \in \mathcal{Y}$ is not learned directly, but in a more indirect way (Pillai et al., 2017; Waegeman et al., 2014; Ye et al., 2012). More specifically, the training data is used to learn a probabilistic predictor, which, given a query instance $\boldsymbol{x}$, predicts a probability distribution $\boldsymbol{p}(\cdot \mid \boldsymbol{x})$ on the set of labelings $\mathcal{Y}$. The Bayes-optimal prediction (BOP) is then given by the expected loss minimizer

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{y}}(\boldsymbol{x}) \in \operatorname*{argmin}_{\bar{\boldsymbol{y}} \in \mathcal{Y}} \mathbf{E}\big(\ell(\boldsymbol{y}, \bar{\boldsymbol{y}})\big) = \operatorname*{argmin}_{\bar{\boldsymbol{y}} \in \mathcal{Y}} \sum_{\boldsymbol{y} \in \mathcal{Y}} \ell(\boldsymbol{y}, \bar{\boldsymbol{y}})\, \boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x})\,. \tag{6}$$

Note that the BOP (6) is defined in a pointwise way, i.e., it is computed for each $\boldsymbol{x}$ individually. Theoretically, a (global) hypothesis $\boldsymbol{h} : \mathcal{X} \longrightarrow \mathcal{Y}$ can of course be constructed by setting $\boldsymbol{h}(\boldsymbol{x}) := \hat{\boldsymbol{y}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$, although the mapping thus obtained is not guaranteed to have any specific structure (i.e., it is normally not an element of a "simple" hypothesis space $\mathcal{H}$).

2.2 MLC Loss Functions

In the literature, various loss functions have been proposed for multilabel classification. Simple (though commonly used) examples are the Hamming loss

$$\ell_H(\boldsymbol{y}, \hat{\boldsymbol{y}}) := \sum_{k=1}^{K} [\![y_k \neq \hat{y}_k]\!] \ , \tag{7}$$

and the *subset 0/1 loss*

$$\ell_S(\boldsymbol{y}, \hat{\boldsymbol{y}}) := [\![\boldsymbol{y} \neq \hat{\boldsymbol{y}}]\!]\,. \tag{8}$$

Both losses generalize the standard $0/1$ loss in binary classification, albeit in a very different way: While Hamming counts the number of labels on which a prediction is wrong, subset $0/1$ is an "all or nothing" loss that merely checks whether the entire label combination is predicted correctly or not.

Another commonly used loss function (or actually utility function) is the F-measure, i.e., the harmonic mean of precision and recall, which is well-known from information retrieval (Decubber et al., 2018; Lewis, 1995). More specifically, the f_β-measure is computed as

$$f_\beta(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{(1+\beta^2)\sum_{k=1}^{K} \hat{y}_k\, y_k}{\sum_{k=1}^{K} \hat{y}_k + \beta^2 \sum_{k=1}^{K} y_k}\,, \tag{9}$$

where β is a parameter that controls the influence of precision and recall; a common choice is $\beta = 1$, giving the same weight to both. The F-Measure is specifically motivated by the strong imbalance between positive (relevant) and negative (irrelevant) labels that is quite common for multilabel data: the number of positive labels is often very small compared to the overall number of labels. As a consequence, even the (uninformed) "all negative" prediction $\hat{\boldsymbol{y}} = \mathbf{0}$ may have a very small Hamming loss. To perform well in terms of the F-measure, however, being correct on the negative labels is not enough; instead, the predictor has to be correct on the positive labels, too.

The F-measure is an example of the so-called confusion matrix-derived performance (CMDP) measures (Luque et al., 2019a,b; Powers, 2011), which can be expressed as functions of the four primitives of the confusion matrix, the true positives (*tp*), false positives (*fp*), false negatives (*fn*), and true negatives (*tn*):

$$
\begin{aligned}
tp &= \sum_{k=1}^{K} y_k \hat{y}_k \,, & fn &= \sum_{k=1}^{K} y_k (1 - \hat{y}_k) \,, \\
fp &= \sum_{k=1}^{K} (1 - y_k) \hat{y}_k \,, & tn &= \sum_{k=1}^{K} (1 - y_k)(1 - \hat{y}_k) \,.
\end{aligned} \tag{10}
$$

Table 2 lists a number of important special cases of such measures, i.e., functions of the form

$$
f \,:\, \mathbb{N}_0^4 \longrightarrow [0, 1] \,. \tag{11}
$$

Note that all measures in Table 2 are expressed as accuracy ("higher is better") measures. However, because they are all normalized and assume values in the unit interval, they can easily be turned into associated loss functions ℓ_f by setting $\ell_f(\boldsymbol{y}, \hat{\boldsymbol{y}}) \coloneqq 1 - f(\boldsymbol{y}, \hat{\boldsymbol{y}})$.

#	Measure	Definition
1	F-measure	$f_\beta = \frac{(1+\beta^2)tp}{(1+\beta^2)tp+\beta^2 fn+fp}$
2	Recall/sensitivity	$f_{\text{Rec}} = \frac{tp}{tp+fn}$
3	Specificity	$f_{\text{Spe}} = \frac{tn}{tn+fp}$
4	Precision	$f_{\text{Pre}} = \frac{tp}{tp+fp}$
5	Negative predictive value	$f_{\text{Neg}} = \frac{tn}{tn+fn}$
6	Jaccard index	$f_{\text{Jac}} = \frac{tp}{tp+fn+fp}$
7	Geometric mean	$f_{\text{Geo}} = \sqrt{f_{\text{Rec}} \cdot f_{\text{Spe}}}$
8	Informedness	$f_{\text{Inf}} = (f_{\text{Spe}}+f_{\text{Rec}})/2$
9	Markedness	$f_{\text{Mar}} = (f_{\text{Neg}}+f_{\text{Pre}})/2$

Table 2: Commonly used confusion matrix-derived accuracy measures

An alternative to predicting label subsets $\hat{\boldsymbol{y}} \in \mathcal{Y}$ is to make predictions in the form a *ranking* of the labels λ_k, that is, a total order specified by a permutation π of $[K] \coloneqq \{1, \ldots, K\}$ such that $\pi(i)$ is the index of the label on position i of the ranking, and $\pi^{-1}(j)$ the position of the j^{th} label λ_j. Thus, a permutation π encodes the ranking

$$
\lambda_{\pi(1)} \succ \lambda_{\pi(2)} \succ \cdots \succ \lambda_{\pi(K)} \,. \tag{12}
$$

Typically, such rankings are obtained by sorting the labels in decreasing order according to their (predicted) probabilities $p_k = \boldsymbol{p}_k(1 \mid \boldsymbol{x})$, i.e., π is such that $\pi^{-1}(i) > \pi^{-1}(j)$ implies $p_i \leq p_j$. The *rank loss* then counts the number of incorrectly ordered label-pairs, that is, the number of pairs λ_i, λ_j such that λ_i is ranked worse than λ_j although λ_i is relevant while λ_j is irrelevant:

$$
\ell_R(\boldsymbol{y}, \pi) = \sum_{(i,j)\,:\, y_i > y_j} \llbracket \pi^{-1}(i) > \pi^{-1}(j) \rrbracket \;,
$$

or equivalently,

$$
\ell_R(\boldsymbol{y}, \pi) = \sum_{1 \leq i < j \leq K} \llbracket y_{\pi(i)} = 0 \wedge y_{\pi(j)} = 1 \rrbracket \,.
$$

2.3 Label Dependence

The goal of classification algorithms in general is to capture dependencies between input features and the target variable. In MLC, dependencies may not only exist between the features and each target, but also between the targets $Y_1, \dots, Y_K$ themselves. The idea to improve predictive accuracy by capturing such dependencies is a driving force in research on multilabel classification.

In this regard, a distinction between *unconditional* and *conditional independence* between labels can be made (Dembczyński et al., 2012). In the first case, the joint distribution $\boldsymbol{p}(\mathbf{Y})$ in the labeling space factorizes into the product of the marginals $\boldsymbol{p}(Y_k)$, i.e.,

$$\boldsymbol{p}(\mathbf{Y}) = \boldsymbol{p}_1(Y_1) \times \boldsymbol{p}_2(Y_2) \times \cdots \times \boldsymbol{p}_K(Y_K)\,, \tag{13}$$

whereas in the latter case, the factorization

$$\boldsymbol{p}(\mathbf{Y} \,|\, \boldsymbol{x}) = \boldsymbol{p}_1(Y_1 \,|\, \boldsymbol{x}) \times \boldsymbol{p}_2(Y_2 \,|\, \boldsymbol{x}) \times \cdots \times \boldsymbol{p}_K(Y_K \,|\, \boldsymbol{x}) \tag{14}$$

holds conditioned on $\boldsymbol{x}$, for every instance $\boldsymbol{x}$. Equivalently, the property of conditional label independence (CLI) can also be expressed as follows:

$$\boldsymbol{p}(\boldsymbol{y} \,|\, \boldsymbol{x}) = \prod_{k=1}^{K} p_k^{y_k}(1 - p_k)^{1-y_k}\,, \tag{15}$$

where $p_k = \boldsymbol{p}_k(1 \,|\, \boldsymbol{x})$ denotes the marginal relevance probability of label λ_k. Thus, unconditional independence is a kind of global independence, whereas conditional independence is an independence locally restricted to a single point in the instance space. If the equality in (13) does not hold (for example due to a hierarchical dependence structure on the label space), we also speak of label dependence. Likewise, a case of conditional label dependence is a case where (14) is violated.

As it turns out, there is a close connection between label dependence and the decomposability of loss functions: A decomposable loss can be expressed in the form

$$\ell(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{k=1}^{K} \ell_k(y_k, \hat{y}_k) \tag{16}$$

with suitable binary loss functions $\ell_k : \{0,1\}^2 \longrightarrow \mathbb{R}$, whereas a non-decomposable loss does not permit an additive representation of that kind. It can be shown that, to produce optimal predictions $\hat{\boldsymbol{y}} = \boldsymbol{h}(\boldsymbol{x})$ minimizing expected loss (6), knowledge about the marginals $\boldsymbol{p}_k(Y_k \,|\, \boldsymbol{x})$ is enough in the case of a decomposable loss (such as Hamming), but not in the case of a non-decomposable loss (Dembczyński et al., 2012). Instead, if a loss is non-decomposable, higher-order probabilities are needed, and in the extreme case even the entire distribution $\boldsymbol{p}(\mathbf{Y} \,|\, \boldsymbol{x})$ (like in the case of the subset $0/1$ loss). For example, one easily verifies that the BOP (6) for the subset $0/1$ loss is given by the *joint* mode of the distribution $\boldsymbol{p}(\cdot \,|\, \boldsymbol{x})$, i.e.,

$$\hat{\boldsymbol{y}} \in \operatorname*{argmax}_{\bar{\boldsymbol{y}} \in \mathcal{Y}} \boldsymbol{p}(\bar{\boldsymbol{y}} \,|\, \boldsymbol{x})\,, \tag{17}$$

whereas the optimal prediction for the Hamming loss is given by the *marginal* mode $(\hat{y}_1, \dots, \hat{y}_K)$ with

$$\hat{y}_k \in \operatorname*{argmax}_{\bar{y}_k \in \{0,1\}} \boldsymbol{p}(\bar{y}_k \,|\, \boldsymbol{x})\,. \tag{18}$$

On an algorithmic level, this means that MLC with a decomposable loss can be tackled by what is commonly called binary relevance (BR) learning (i.e., learning one binary classifier for each label individually), whereas non-decomposable losses call for more sophisticated learning methods that are able to take label-dependencies into account.

Example 1. *Consider a multilabel classification problem with three labels, and suppose the following conditional distribution on labelings* $\boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x})$ *to be given (with positive probability for five labelings and 0 for all others):*

$\boldsymbol{y} = (y_1, y_2, y_3)$	$p(\boldsymbol{y} \mid \boldsymbol{x})$
$(0, 0, 0)$	$1/4$
$(1, 1, 1)$	$3/16$
$(0, 1, 1)$	$3/16$
$(1, 0, 1)$	$3/16$
$(1, 1, 0)$	$3/16$

According to (17), the BOP for the subset $0/1$ *loss is given by* $\hat{\boldsymbol{y}} = (0, 0, 0)$*, while the BOP for the Hamming loss is* $\hat{\boldsymbol{y}} = (1, 1, 1)$ *according to (18).*

3. MLC with Partial Abstention

In our generalized setting of MLC with abstention, which is introduced in this section, the classifier is allowed to produce *partial predictions*

$$\hat{\boldsymbol{y}} = \mathbf{h}(\boldsymbol{x}) \in \mathcal{Y}^* := \{0, \perp, 1\}^K , \tag{19}$$

where $\hat{y}_k = \perp$ indicates an abstention on the label λ_k; we denote by

$$A(\hat{\boldsymbol{y}}) \subseteq [K] := \{1, \ldots, K\} \quad \text{and} \quad D(\hat{\boldsymbol{y}}) := [K] \setminus A(\hat{\boldsymbol{y}})$$

the set of indices k for which $\hat{y}_k = \perp$ and $\hat{y}_k \in \{0, 1\}$, respectively, that is, the indices on which the learner abstains and decides to predict.

Note that a partial prediction $\hat{\boldsymbol{y}}$ can be associated with a set-valued prediction $\hat{Y} \subseteq \mathcal{Y}$, namely the set of consistent instantiations (extensions) of $\hat{\boldsymbol{y}}$:

$$\hat{Y} = \left\{ \boldsymbol{y} \in \mathcal{Y} \mid \forall i \in D(\hat{\boldsymbol{y}}) : y_i = \hat{y}_i \right\} . \tag{20}$$

Thus, we can look at a partial prediction as both an element of $\mathcal{Y}^*$ (a vector with entries 0, 1, and $\perp$) and a subset of $\mathcal{Y}$. The set (20) can be seen as a kind of confidence set, namely a set of candidate labelings that is supposed to cover the ground-truth labeling $\boldsymbol{y}$. This reflects a main motivation for an MLC classifier to (partly) abstain, namely to guarantee its "reliability" in cases of uncertainty.

Although we will not pursue this direction further in this paper, let us note that, in principle, one may even allow the learner to predict *any* subset $\hat{Y} \subseteq \mathcal{Y}$, not only subsets that have a representation (20). This might be important, for example, in the case of dependencies between labels. For instance, the learner may wish to express that the labels λ_i and λ_j are either both relevant or both irrelevant (i.e., y_i and y_j are either both 0 or both 1). Obviously, while a partial prediction (20) is a very compact representation of a subset $\hat{Y} \subseteq \mathcal{Y}$ in terms of a vector $\hat{\boldsymbol{y}}$, the representation of arbitrary subsets is a non-trivial problem, because a simple enumeration will not be feasible in general.

3.1 Risk Minimization

To evaluate a partially abstaining multilabel classifier, a generalized MLC loss function

$$L : \mathcal{Y} \times \mathcal{Y}^* \longrightarrow \mathbb{R}_+ \tag{21}$$

is needed, which compares a partial prediction $\hat{\boldsymbol{y}}$ with a ground-truth labeling $\boldsymbol{y}$. Given a loss of that kind, and assuming a probabilistic prediction for $\boldsymbol{x}$, i.e., a probability $\boldsymbol{p}(\cdot \mid \boldsymbol{x})$ on the set of labelings $\mathcal{Y}$, the problem of finding the BOP (minimizing expected loss) comes down to finding

$$\hat{\boldsymbol{y}} \in \operatorname*{argmin}_{\bar{\boldsymbol{y}} \in \mathcal{Y}^*} \mathbf{E}\big(L(\boldsymbol{y}, \bar{\boldsymbol{y}})\big) = \operatorname*{argmin}_{\bar{\boldsymbol{y}} \in \mathcal{Y}^*} \sum_{\boldsymbol{y} \in \mathcal{Y}} L(\boldsymbol{y}, \bar{\boldsymbol{y}}) \cdot \boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x}) \,. \tag{22}$$

The concrete form of this optimization problem as well as its difficulty depend on several choices, including the underlying MLC loss function ℓ and its extension L. It is clear that explicitly solving (22) as a combinatorial optimization problem is even more challenging than solving the optimization problem in the standard setting of MLC (6), since the expectation needs to be computed over $|\mathcal{Y}^*| = 3^K$ instead of 2^K candidate predictions.

3.2 Generalized Loss Functions

On the basis of a standard MLC loss ℓ, a generalized loss function (21) can be derived in different ways, also depending on how to penalize the abstention. Further below, we propose a generalization based on an additive penalty. Before doing so, we discuss some general properties that might be of interest for generalized losses.

3.2.1 PROPERTIES OF GENERALIZED MLC LOSSES

As a first property, we should expect a generalized loss L to reduce to its conventional version ℓ in the case of no abstention. In other words,

$$L(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \ell(\boldsymbol{y}, \hat{\boldsymbol{y}})$$

whenever $\hat{\boldsymbol{y}}$ is a complete prediction, i.e., an element of $\mathcal{Y}$. Needless to say, this is a property that every generalized loss should obey.

Another reasonable property is *monotonicity*, which requires an appropriate "appreciation" of abstention. More specifically, it appears natural to require that the loss should only increase (or at least not decrease) in any of the following cases:

(D1) turning a correct prediction on a label λ_k into an incorrect prediction,

(D2) turning a correct prediction on a label λ_k into an abstention,

(D3) turning an abstention into an incorrect prediction.

Definition 1 (Monotonicity)**.** *A generalized loss L is monotonic if it only increases (or at least not decreases) in any of the cases D1, D2, D3.*

This reflects the following chain of preferences: a correct prediction is better than an abstention, which in turn is better than an incorrect prediction. More formally, for a ground-truth labeling $\boldsymbol{y}$ and a partial prediction $\hat{\boldsymbol{y}}_1$, let $C_1, A_1 \subseteq \mathcal{L}$ denote the subset of labels on which the prediction is correct and on which the learner abstains, respectively, and define $C_2, A_2 \subseteq \mathcal{L}$ analogously for a prediction $\hat{\boldsymbol{y}}_2$. Then

$$(C_2 \subseteq C_1) \ \wedge \ \Big((C_2 \cup A_2) \subseteq (C_1 \cup A_1)\Big) \Rightarrow L(\boldsymbol{y}, \hat{\boldsymbol{y}}_1) \leq L(\boldsymbol{y}, \hat{\boldsymbol{y}}_2) \, . \tag{23}$$

Yet another interesting property is what we call *uncertainty-alignment*. Intuitively, when producing a partial prediction, an optimal prediction rule is supposed to abstain on the most uncertain labels.

Definition 2 (Uncertainty-Aligned)**.** *Consider a generalized loss function L and a prediction $\hat{\boldsymbol{y}}$ which, for a query $\boldsymbol{x} \in \mathcal{X}$, is a risk-minimizer (22). Moreover, denoting by $p_k = \boldsymbol{p}_k(1 \mid \boldsymbol{x})$ the (marginal) probability that label λ_k is relevant for $\boldsymbol{x}$, it is natural to quantify the degree of uncertainty on this label in terms of*

$$u_k = 1 - 2|p_k - 1/2| = 2\min(p_k, 1 - p_k) \, , \tag{24}$$

or any other function symmetric around $1/2$. We say that $\hat{\boldsymbol{y}}$ is uncertainty-aligned *if*

$$\forall\, i \in A(\hat{\boldsymbol{y}}), j \in D(\hat{\boldsymbol{y}}) \, : \ u_i \geq u_j \, . \tag{25}$$

Thus, a prediction is uncertainty-aligned if the following holds: Whenever the learner decides to abstain on label λ_i and to predict on label λ_j, the uncertainty on λ_j cannot exceed the uncertainty on λ_i. Or, stated differently, if a learner abstains on λ_i, and λ_j is a label on which it is even more uncertain, then it should also abstain on λ_j. We then call a loss function L uncertainty-aligned if it admits an uncertainty-aligned BOP, i.e., if it guarantees the existence of an uncertainty-aligned risk-minimizer, regardless of the probability $\boldsymbol{p} = \boldsymbol{p}(\cdot \mid \boldsymbol{x})$.

As a relaxation of uncertainty-alignment, we further introduce the notion of semi-uncertainty-alignment.

Definition 3 (Semi-Uncertainty-Aligned)**.** *A prediction is semi-uncertainty-aligned if the following generalization of (25) holds:*

$$\exists\, \alpha \in [0,1] \,\forall\, i \in A(\hat{\boldsymbol{y}}), j \in D(\hat{\boldsymbol{y}}) \, : \ |p_i - \alpha| \leq |p_j - \alpha| \, . \tag{26}$$

Obviously, (25) is recovered as a special case for $\alpha = 1/2$. In (26), the reference point α can be any value in the unit interval. This value is not supposed to be known, only to exist, and may also depend on the underlying probability distribution $\boldsymbol{p}$. Note that (26) is equivalent to the following condition: After sorting the labels according to their marginal probabilities, the prediction set is obtained by abstaining on the "middle part". More precisely, suppose the labels are sorted in decreasing order of their marginal probabilities p_k, resulting in a ranking π of the form (12). Then, a prediction $\hat{\boldsymbol{y}}$ is of the form

$$\hat{y}_k = \begin{cases} 1 & \text{if } \pi^{-1}(k) \leq l \\ \bot & \text{if } l < \pi^{-1}(k) < r \\ 0 & \text{if } r \leq \pi^{-1}(k) \end{cases} ,$$

where $l, r \in \{0, 1, \ldots, K+1\}$ are indices such that $l < r$. In other words, a prediction is specified by a decision set

$$D(\hat{\boldsymbol{y}}) = \langle\!\langle l, r \rangle\!\rangle := \{1, \ldots, l\} \cup \{r, \ldots, K\} \tag{27}$$

in the form of a union of “boundary positions”. Again, we call a loss function L semi-uncertainty-aligned if it admits a semi-uncertainty-aligned BOP, i.e., if it guarantees the existence of a semi-uncertainty-aligned risk-minimizer with a decision set of the form (27), regardless of the underlying probability $\boldsymbol{p} = \boldsymbol{p}(\cdot \mid \boldsymbol{x})$.

3.2.2 ADDITIVE PENALTY FOR ABSTENTION

Consider the case of a partial prediction $\hat{\boldsymbol{y}}$ and denote by $\hat{\boldsymbol{y}}_D$ the projections of $\hat{\boldsymbol{y}}$ to the entries in $D(\hat{\boldsymbol{y}})$. As a natural extension of the original loss ℓ, we propose a generalized loss of the form

$$L(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \ell(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) + g(A(\hat{\boldsymbol{y}}))\,, \tag{28}$$

with $\ell(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)$ the original loss on that part on which the learner predicts and $g(A(\hat{\boldsymbol{y}}))$ a penalty for abstaining on the label subset $A(\hat{\boldsymbol{y}})$. The latter can be seen as a measure of the loss of usefulness of the prediction $\hat{\boldsymbol{y}}$ due to its partiality, i.e., due to having no predictions on $A(\hat{\boldsymbol{y}})$.

An important instantiation of (28) is the case where the penalty is a counting measure, i.e., where g only depends on the number of abstentions:

$$L(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \ell(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) + g\big(|A(\hat{\boldsymbol{y}})|\big)\,. \tag{29}$$

A special case of (29) in turn is to penalize each abstention $\hat{y}_k = \perp$ with the same constant $c \in [0, 1]$, which yields

$$L(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \ell(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) + |A(\hat{\boldsymbol{y}})| \cdot c\,. \tag{30}$$

Of course, instead of a linear function g, more general penalty functions are conceivable. For example, a practically relevant penalty is a concave function of the number of abstentions: Each additional abstention causes additional cost, so g is monotone increasing in $|A(\hat{\boldsymbol{y}})|$, but the marginal cost of abstention is decreasing. An example of a concave penalty function (that we shall use later on) is

$$L(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \ell(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) + \frac{|A(\hat{\boldsymbol{y}})| \cdot K \cdot c}{K + |A(\hat{\boldsymbol{y}})|}\,. \tag{31}$$

3.3 Inferring Bayes-Optimal Predictions

Adhering to the decision-theoretic approach, and assuming probabilities $\boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x})$ to be made available by an MLC predictor $\boldsymbol{h}$ for a given query instance $\boldsymbol{x}$, our main interest in the setting of MLC with partial abstention is finding Bayes-optimal predictions, i.e., predictions minimizing a generalized MLC loss L in expectation. To facilitate the readability of the technical exposition that will follow in the subsequent sections, let us provide a short outline and summary of the main contents of these sections:

- The generalization (28) is generic and can be used to extend any MLC loss function to the setting of MLC with partial abstention. As a first attempt, we shall focus on the natural case (29), where the penalty only depends on the number of labels on which the learner abstains.

- We show that, for different types of MLC loss functions, the Bayes-optimal prediction for the generalization (29) is semi-uncertainty-aligned or even uncertainty-aligned. As already said, this property implies a specific structure of an optimal prediction, namely a prediction set of the form (27), which in turn allows for finding a BOP in an efficient way: Instead of checking the entire (exponentially large) set of candidate labelings, one only needs to find the right "middle part", i.e., the indices l and r in (27), after sorting the labels λ_k according to their marginal probabilities p_k. Therefore, the number of candidates reduces from exponential to quadratic.

- The case of uncertainty-alignment, with known reference point $\alpha = 1/2$, admits an even more efficient approach: After sorting the labels λ_k in increasing order according to their calibrated uncertainties $|p_k - 1/2|$, the abstention set corresponds to the top of the sorted list, and can hence be found in linear time.
- We start with the case of decomposable losses (Section 4), for which we show that a BOP can be found efficiently, regardless of whether the labels are (conditionally) independent or not.
- For the case of non-decomposable losses, we have to make an assumption of CLI (15). Under this assumption, we show that the optimal predictions for the generalization (29) of the rank loss (Section 5), the subset 0/1 loss (Section 6), and a family of CMDP measures (Section 7) are all semi-uncertainty aligned. As explained above, this allows us to devise efficient algorithms.

4. The Case of Decomposable Losses

We start with the general case of decomposable losses in the sense of (16), where $\ell_k(0,1)$ and $\ell_k(1,0)$ are not necessarily equal. For any $d = 0, \ldots, K$, denote by

$$\mathcal{Y}_d^* := \left\{ \bar{\boldsymbol{y}} \in \mathcal{Y}^* \mid |D(\bar{\boldsymbol{y}})| = d \right\} \tag{32}$$

the set of labelings with exactly d decided (positive or negative) labels and $K - d$ abstentions. We can decompose the optimization problem (22) into an inner and an outer minimization task as follows:

$$\hat{\boldsymbol{y}}^d := \operatorname*{argmin}_{\bar{\boldsymbol{y}} \in \mathcal{Y}_d^*} \mathbf{E}\left(L(\boldsymbol{y}, \bar{\boldsymbol{y}})\right) , \tag{33}$$

$$\hat{\boldsymbol{y}} := \operatorname*{argmin}_{\bar{\boldsymbol{y}} \in \{\hat{\boldsymbol{y}}^0, \ldots, \hat{\boldsymbol{y}}^K\}} \mathbf{E}\left(L(\boldsymbol{y}, \bar{\boldsymbol{y}})\right) . \tag{34}$$

Proposition 1. *Let π be a permutation that sorts the labels in increasing order of the label-wise expected loss*

$$\begin{aligned} s_k &:= \min_{\bar{y}_k \in \{0,1\}} \mathbf{E}(\ell_k(y_k, \bar{y}_k)) = \min_{\bar{y}_k \in \{0,1\}} \sum_{\boldsymbol{y} \in \mathcal{Y}} \ell_k(y_k, \bar{y}_k) \boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x}) \\ &= \min_{\bar{y}_k \in \{0,1\}} \ell_k(1 - \bar{y}_k, \bar{y}_k)(p_k)^{1-\bar{y}_k}(1 - p_k)^{\bar{y}_k} \\ &= \min \left\{ (1 - p_k)\ell_k(0,1), p_k \ell_k(1,0) \right\} , \end{aligned}$$

i.e., the permutation π is such that

$$s_{\pi(1)} \le s_{\pi(2)} \le \ldots \le s_{\pi(K)} . \tag{35}$$

If the loss ℓ is decomposable in the sense of (16)*, a BOP*

$$\hat{\boldsymbol{y}} \in \operatorname*{argmin}_{\bar{\boldsymbol{y}} \in \mathcal{Y}^*} \mathbf{E}\left(L(\boldsymbol{y}, \bar{\boldsymbol{y}})\right) = \operatorname*{argmin}_{0 \le d \le K} \mathbf{E}\left(\ell(\boldsymbol{y}, \hat{\boldsymbol{y}}^d)\right) + g(K - d) \tag{36}$$

of the generalization (29) *is given by the partial prediction* $\hat{\boldsymbol{y}}^d$ *with*

$$\hat{y}^d_k = \operatorname*{argmin}_{\bar{y}_k \in \{0,1\}} \mathbf{E}\left(\ell_k(y_k, \bar{y}_k)\right) , \tag{37}$$

on the index set $D(\hat{\boldsymbol{y}}^d) = \{k \in [K] \,|\, \pi^{-1}(k) \leq d\}$.

As shown by the previous proposition, a BOP for a decomposable loss can easily be found in time $O(K \log(K))$, which is the time needed to obtain the sorting (35) of the labels according to the scores s_k. Given this sorting, the optimal prediction of size d can be found without further searching, as it is always given by the d labels with lowest scores. Therefore, the optimal prediction set along with the prediction itself can be produced in linear time, simply by finding the optimal size d of the prediction set according to (36), i.e., by trying each set size d and picking the best.

As a consequence of Proposition 1, we obtain the following result for the generalized Hamming loss (29) .

Corollary 1. *Let* π *be a permutation that sorts the labels in increasing order of the degree of uncertainty* (24)*. In the case of the generalized Hamming loss* (29)*, a BOP*

$$\hat{\boldsymbol{y}} \in \operatorname*{argmin}_{\bar{\boldsymbol{y}} \in \mathcal{Y}^*} \mathbf{E}\left(L_H(\boldsymbol{y}, \bar{\boldsymbol{y}})\right) = \operatorname*{argmin}_{0 \leq d \leq K} \mathbf{E}\left(\ell_H(\boldsymbol{y}, \hat{\boldsymbol{y}}^d)\right) + g(K-d) \tag{38}$$

is given by the prediction $\hat{\boldsymbol{y}}^d$ *with*

$$\hat{y}^d_k = \operatorname*{argmin}_{\bar{y}_k \in \{0,1\}} \mathbf{E}\left(\ell_k(y_k, \bar{y}_k)\right) = \operatorname*{argmin}_{\bar{y}_k \in \{0,1\}} (p_k)^{1-\bar{y}_k}(1-p_k)^{\bar{y}_k} \tag{39}$$

on the index set $D(\hat{\boldsymbol{y}}^d) = \{k \in [K] \,|\, \pi^{-1}(k) \leq d\}$*. This prediction can be found in time* $O(K \log(K))$.

Corollary 2. *The generalized Hamming loss* (29) *is uncertainty-aligned. In the case of the generalized Hamming loss* (30)*, the BOP is given by* (39) *with*

$$D(\hat{\boldsymbol{y}}^d) = \{k \in [K] \,|\, \min\left\{p_k, 1-p_k\right\} \leq c\} .$$

Thus, a BOP of the generalized Hamming loss (30) can easily be found in time $O(K)$, simply by comparing $\min\{p_k, 1-p_k\}$ to the cost value c.

Remark 1. *The generalized Hamming loss* (29) *is monotonic, provided* g *is non-decreasing and such that* $g(k+1) - g(k) \leq 1$, $\forall\, k \in [K-1]$.

5. The Case of Rank Loss

As already said, in the case of the rank loss, we assume predictions in the form of rankings (12) instead of labelings. The rank loss then counts the number of incorrectly ordered label-pairs, that is, the number of pairs λ_i, λ_j such that λ_i is ranked worse than λ_j although λ_i is relevant while λ_j is irrelevant:

$$\ell_R(\boldsymbol{y}, \pi) = \sum_{1 \leq i < j \leq K} [\![y_{\pi(i)} = 0 \wedge y_{\pi(j)} = 1]\!] . \tag{40}$$

Thus, given that the ground-truth labeling is distributed according to the probability $\boldsymbol{p}(\cdot \mid \boldsymbol{x})$, the expected loss of a predicted ranking π is

$$\mathbf{E}\left(\ell_R(\boldsymbol{y}, \pi)\right) = \sum_{\boldsymbol{y} \in \mathcal{Y}} \ell_R(\boldsymbol{y}, \pi) \boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{1 \leq i < j \leq K} \boldsymbol{p}_{\pi(i),\pi(j)}(0, 1 \mid \boldsymbol{x}) , \tag{41}$$

where $\boldsymbol{p}_{u,v}$ denotes the pairwise marginal probabilities

$$\boldsymbol{p}_{u,v}(a, b) = \boldsymbol{p}_{u,v}(a, b \mid \boldsymbol{x}) = \sum_{\boldsymbol{y} \in \mathcal{Y}:\, y_u = a, y_v = b} \boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x}) . \tag{42}$$

In the following, we first recall the BOP for the rank loss as introduced above and then generalize it to the case of partial predictions. We use the following notation: For a labeling $\boldsymbol{y}$, let $s(\boldsymbol{y}) = \sum_{k=1}^{K} y_k$ be the number of relevant labels, and $c(\boldsymbol{y}) = s(\boldsymbol{y})(K - s(\boldsymbol{y}))$ the number of relevant/irrelevant label pairs (and hence an upper bound on the rank loss).

As shown by Dembczyński et al. (2012), a BOP ranking π, i.e., a ranking minimizing (41), is provably obtained by sorting the labels λ_k in decreasing order of the marginal probabilities p_k, i.e., according to their probability of being relevant. Thus, a BOP π is such that

$$p_{\pi(1)} \geq p_{\pi(2)} \geq \ldots \geq p_{\pi(K)} . \tag{43}$$

To show this result, let $\bar{\pi}$ denote the reversal of π, i.e., the ranking that reverses the order of the labels. Then, for each pair (i, j) such that $y_i > y_j$, either π or $\bar{\pi}$ incurs an error, but not both. Therefore, $c(\boldsymbol{y}) = \ell_R(\boldsymbol{y}, \pi) + \ell_R(\boldsymbol{y}, \bar{\pi})$, and

$$\ell_R(\boldsymbol{y}, \pi) - \ell_R(\boldsymbol{y}, \bar{\pi}) = 2\ell_R(\boldsymbol{y}, \pi) - c(\boldsymbol{y}) . \tag{44}$$

Because $c(\boldsymbol{y})$ is a constant that does not depend on π, minimizing $\ell_R(\boldsymbol{y}, \pi)$ (in expectation) is equivalent to minimizing the difference $\ell_R(\boldsymbol{y}, \pi) - \ell_R(\boldsymbol{y}, \bar{\pi})$. For the latter, the expectation (41) becomes

$$\begin{aligned} \mathbf{E}\left(\ell_R(\boldsymbol{y}, \pi) - \ell_R(\boldsymbol{y}, \bar{\pi})\right) &= \sum_{\boldsymbol{y} \in \mathcal{Y}} \left(\ell_R(\boldsymbol{y}, \pi) - \ell_R(\boldsymbol{y}, \bar{\pi})\right) \boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x}) \\ &= \sum_{1 \leq i < j \leq K} \left(\boldsymbol{p}_{\pi(i),\pi(j)}(0, 1) - \boldsymbol{p}_{\pi(i),\pi(j)}(1, 0)\right) \\ &= \sum_{1 \leq i < j \leq K} \left(p_{\pi(j)} - p_{\pi(i)}\right) = \sum_{1 \leq i \leq K} (2k - (K + 1))\, p_{\pi(k)} , \end{aligned} \tag{45}$$

where the transition from the first to the second sum is valid because

$$\begin{aligned} \boldsymbol{p}_{u,v}(0, 1) - \boldsymbol{p}_{u,v}(1, 0) &= \boldsymbol{p}_{u,v}(0, 1) + \boldsymbol{p}_{u,v}(1, 1) - \boldsymbol{p}_{u,v}(1, 1) - \boldsymbol{p}_{u,v}(1, 0) \\ &= \boldsymbol{p}_v(1) - \boldsymbol{p}_u(1) = p_v - p_u . \end{aligned}$$

From (45), it is clear that a BOP ranking π is defined by (43).

To generalize this result, let us look at the rank loss of a partial prediction of fixed size $d \in [K]$, i.e., a ranking of a subset of d labels. To simplify notation, we identify such a prediction, not with the original set of indices of the labels, but the positions of the corresponding labels in the sorting (43). Thus, a partial prediction of size d is identified by a set of indices $\boldsymbol{D}_d = \{k_1, \ldots, k_d\}$ such that $k_1 < k_2 < \ldots < k_d$, where $k \in \boldsymbol{D}_d$ means that the label $\lambda_{\pi(k)}$ with the k^{th} largest probability

$p_{\pi(k)}$ in (43) is included. According to the above result, the optimal ranking π_{D_d} on these labels is the identity, and the expected loss of this ranking is given by

$$\mathbf{E}\left(\ell_R(\boldsymbol{y}, \pi_{D_d})\right) = \sum_{1 \le i < j \le d} \boldsymbol{p}_{\pi(k_i),\pi(k_j)}(0,1)\,. \tag{46}$$

Lemma 1. *Assuming CLI in the sense that $\boldsymbol{p}_{i,j}(y_i, y_j) = p_i p_j$, the generalized rank loss* (29) *is semi-uncertainty-aligned. Thus, the Bayes-optimal prediction is a partial prediction with decision set of the form*

$$D_d = D_d(\hat{\boldsymbol{y}}) = \langle\!\langle l, r \rangle\!\rangle := \{1, \dots, l\} \cup \{r, \dots, K\}\,, \tag{47}$$

where $0 \le l < r \le K + 1$ *(and* $d = K - r + l + 1$*).*

According to the previous lemma, an optimal d-selection D_d leading to an optimal (partial) ranking of length d is always a "boundary set" of positions in the ranking (43). The next lemma establishes an important relationship between optimal selections of increasing length.

Lemma 2. *Let $D_d = \langle\!\langle l, r \rangle\!\rangle$ be an optimal d-selection* (47) *for $d \ge 2$. At least one of the extensions $\langle\!\langle l+1, r \rangle\!\rangle$ or $\langle\!\langle l, r-1 \rangle\!\rangle$ of D_d is an optimal* $(d+1)$*-selection.*

Thanks to the previous lemma, a BOP of the generalized rank loss (29) can be constructed quite easily (in time $O(K \log(K))$. First, the labels are sorted according to (43). Then, an optimal decision set is produced by starting with the boundary set $\emptyset$ and increasing this set in a greedy manner.

Proposition 2. *A BOP ranking of the generalized rank loss* (29) *can be constructed in time* $O(K \log(K))$ *using Algorithm 1.*

Algorithm 1 BOP of the generalized rank loss

```
Input: marginal probabilities $(p_1, \dots, p_K) = h(\boldsymbol{x})$, penalty function $g$
Sort $\mathbf{p} := \{p_1, p_2, \dots, p_K\}$ in decreasing order: $p_{\pi(1)} \ge p_{\pi(2)} \ge \dots \ge p_{\pi(K)}$
$D_0 := \emptyset$, $\mathbf{E}_0 := g(K)$
$\mathbf{K}_2 := \langle\!\langle 1, K \rangle\!\rangle$, $l := 1$, $r := K$, $\mathbf{E}_2 = \mathbf{E}(\ell_R(\boldsymbol{y}, \pi_{\mathbf{K}_2})) + g(K-2)$
for $d = 3$ to $K$ do
    $\mathbf{K}_l := \langle\!\langle l+1, r \rangle\!\rangle$, $\mathbf{K}_r := \langle\!\langle l, r-1 \rangle\!\rangle$
    if $\mathbf{E}(\ell_R(\boldsymbol{y}, \pi_{\mathbf{K}_l})) < \mathbf{E}(\ell_R(\boldsymbol{y}, \pi_{\mathbf{K}_r}))$ then
        $D_d := \mathbf{K}_l$, $l := l+1$, $\mathbf{E}_d := \mathbf{E}(\ell_R(\boldsymbol{y}, \pi_{\mathbf{K}_l})) + g(K-d)$
    else
        $D_d := \mathbf{K}_r$, $r := r-1$, $\mathbf{E}_d := \mathbf{E}(\ell_R(\boldsymbol{y}, \pi_{\mathbf{K}_r})) + g(K-d)$
    end if
end for
$d = \mathrm{argmin}_{d \in \{0,2,\dots,K\}} \mathbf{E}_d$
Output: the ranking $\pi_{D_d}$
```

Remark 2. *The generalized rank loss* (29) *is not uncertainty-aligned.*

Because a prediction is a (partial) ranking instead of a (partial) labeling, the property of monotonicity as defined in Section 3.2 does not apply in the case of rank loss. Although it would be possible to generalize this property, for example by looking at (in)correctly sorted label pairs instead of (in)correct labels, we refrain from a closer analysis here.

6. The Case of Subset 0/1 Loss

In the following, we show that under the assumption (15) of conditional label independence, the BOP of the generalized subset $0/1$ loss L_S (29) can be constructed in time $O(K \log(K))$ given the marginal probabilities p_k, $k \in [K]$.

Proposition 3. *Let π be a permutation that sorts the labels in increasing order of the degree of uncertainty* (24)*. In the case of generalized subset* $0/1$ *loss* (29)*, a BOP*

$$\hat{\boldsymbol{y}} \in \operatorname*{argmin}_{\bar{\boldsymbol{y}} \in \mathcal{Y}^*} \mathbf{E}\left(L_S(\boldsymbol{y}, \bar{\boldsymbol{y}})\right) = \operatorname*{argmin}_{0 \leq d \leq K} \mathbf{E}\left(\ell_S(\boldsymbol{y}, \hat{\boldsymbol{y}}^d)\right) + g(K-d) \tag{48}$$

is given by

$$\hat{y}_k^d = \operatorname*{argmin}_{\bar{y}_k \in \{0,1\}} (p_k)^{1-\bar{y}_k}(1-p_k)^{\bar{y}_k} \,, \tag{49}$$

on the index set $D(\hat{\boldsymbol{y}}^d) = \{k \in [K] \mid \pi^{-1}(k) \leq d\}$.

Thus, a BOP for the generalized subset $0/1$ loss (29) can be found in time $O(K \log(K))$, simply by sorting the labels according to the uncertainty degrees u_k, and then finding the optimal size d of the prediction according to (48). Also, the generalized $0/1$ loss (29) is uncertainty-aligned.

7. The Case of Confusion Matrix-Derived Accuracy Measures

As already mentioned, performance measures f derived from the confusion matrix are mostly accuracy measures instead of loss functions. However, because they are typically normalized to the unit interval, they can be turned into loss functions by setting $\ell(\boldsymbol{y}, \hat{\boldsymbol{y}}) = 1 - f(\boldsymbol{y}, \hat{\boldsymbol{y}})$. Thus, the problem of finding a BOP (22) for a generalized loss (29) can be expressed equivalently in terms of the generalization F of the measure f:

$$\begin{aligned} \hat{\boldsymbol{y}} \in \operatorname*{argmax}_{\bar{\boldsymbol{y}} \in \mathcal{Y}^*} \mathbf{E}\big(F(\boldsymbol{y}, \bar{\boldsymbol{y}})\big) &= \operatorname*{argmax}_{\bar{\boldsymbol{y}} \in \mathcal{Y}^*} \sum_{\boldsymbol{y} \in \mathcal{Y}} F(\boldsymbol{y}, \bar{\boldsymbol{y}}) \cdot \boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x}) \\ &= \operatorname*{argmax}_{\bar{\boldsymbol{y}} \in \mathcal{Y}^*} \sum_{\boldsymbol{y} \in \mathcal{Y}} f(\boldsymbol{y}_D, \bar{\boldsymbol{y}}_D) \cdot \boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x}) - g\big(|A(\bar{\boldsymbol{y}})|\big) \,, \end{aligned} \tag{50}$$

where

$$F(\boldsymbol{y}, \bar{\boldsymbol{y}}) = f(\boldsymbol{y}_D, \bar{\boldsymbol{y}}_D) - g\big(|A(\bar{\boldsymbol{y}})|\big) \,. \tag{51}$$

Moreover, the optimization problem (50) can again be decomposed into an inner and an outer maximization as follows:

$$\hat{\boldsymbol{y}}^d := \operatorname*{argmax}_{\bar{\boldsymbol{y}} \in \mathcal{Y}_d^*} \mathbf{E}\left(F(\boldsymbol{y}, \bar{\boldsymbol{y}})\right) \,, \tag{52}$$

$$\hat{\boldsymbol{y}} := \operatorname*{argmax}_{\bar{\boldsymbol{y}} \in \{\hat{\boldsymbol{y}}^0, \ldots, \hat{\boldsymbol{y}}^K\}} \mathbf{E}\left(F(\boldsymbol{y}, \bar{\boldsymbol{y}})\right) \,. \tag{53}$$

Note that many commonly used confusion matrix-derived accuracy measures, including the ones given in Table 2, satisfy the following properties.

Definition 4 (Monotonic Accuracy Measure)**.** *A confusion matrix-derived accuracy measure f is monotonic if it is*

(D4) monotone increasing in tp and tn,

(D5) monotone decreasing in fp and fn.

Thus, in the following, we restrict ourselves to accuracy measures that are monotonic in the sense of Definition 4.

7.1 Monotonicity

From now on, we write $f(tp,fn,fp,tn)$ and $f(\boldsymbol{y},\hat{\boldsymbol{y}})$ exchangeably. Furthermore, we introduce the following shorthand notation:

$$f_S^V(\boldsymbol{y},\hat{\boldsymbol{y}}) = f(tp + v_{tp}, fn + v_{fn}, fp + v_{fp}, tn + v_{tn}), \tag{54}$$

where $S \subseteq \{tp,fn,fp,tn\}$ is the set of quantities that are modified and $V \subseteq \{v_{tp}, v_{fn}, v_{fp}, v_{tn}\}$ is the set of corresponding values added to the original quantities. For example, we write

$$f_{fn,tn}^{-1,+1}(\boldsymbol{y},\hat{\boldsymbol{y}}) = f(tp, fn-1, fp, tn+1).$$

Proposition 4. *Let f be any MLC accuracy measure f of the form* (11) *that is monotonic in the sense of Definition 4. Let g be a non-decreasing function of $|A(\hat{\boldsymbol{y}})|$. The generalized accuracy measure F of the form* (51) *is monotonic in the sense of Definition 1 if and only if*

$$g(|A(\hat{\boldsymbol{y}})|+1) - g(|A(\hat{\boldsymbol{y}})|) \leq \min\left(f_{fp}^{-1}(\boldsymbol{y}_D,\hat{\boldsymbol{y}}_D), f_{fn}^{-1}(\boldsymbol{y}_D,\hat{\boldsymbol{y}}_D)\right) - f(\boldsymbol{y}_D,\hat{\boldsymbol{y}}_D), \tag{55}$$

for any pair $(\boldsymbol{y},\hat{\boldsymbol{y}}) \in \mathcal{Y} \times \mathcal{Y}^*$.

The condition (55) simply means the the reward for turning an incorrect prediction on a label λ_k into an abstention should be at least as high as the cost of doing an extra abstention. In particular, the generalization (51) of an accuracy measure f of the form (11) does not satisfy the condition (55) if g is a strictly monotone increasing function of $|A(\hat{\boldsymbol{y}})|$ and either fp or fn is not taken into account when computing f. Examples include recall/sensitivity, specificity, precision, and negative predictive value. In such a case, there is at least one pair $(\boldsymbol{y},\hat{\boldsymbol{y}}) \in \mathcal{Y} \times \mathcal{Y}^*$ such that

$$f(\boldsymbol{y}_D,\hat{\boldsymbol{y}}_D) = \min\left(f_{fp}^{-1}(\boldsymbol{y}_D,\hat{\boldsymbol{y}}_D), f_{fn}^{-1}(\boldsymbol{y}_D,\hat{\boldsymbol{y}}_D)\right).$$

7.2 General Structure of BOPs

Let π be a permutation that sorts the labels in decreasing order of the marginal probabilities p_k, $k \in [K]$. Assuming CLI in the sense of (15), Lewis (1995) showed that the original f_β has a BOP of the form

$$\hat{y}_k = [\![\pi^{-1}(k) \leq t]\!], k \in [K], \tag{56}$$

where $t \in \{0,1,\ldots,K\}$ is a threshold. The following remark shows that this characterization holds as long as f is an MLC accuracy measure satisfying (D4) and (D5) as given in Definition 4.

Remark 3. *Let π be a permutation that sorts the labels in decreasing order of the marginal probabilites p_k, $k \in [K]$. Assume CLI in the sense of* (15)*, and let f be an MLC accuracy measure of the form* (11) *that is monotonic in the sense of Definition 4. Then, for any $(\hat{\boldsymbol{y}},k) \in \mathcal{Y} \times [K]$, we have the following properties:*

(R1) $\mathbf{E}\left(f(\boldsymbol{y}, \hat{\boldsymbol{y}})\right)$ *is monotone increasing in* p_k *if* $\hat{y}_k = 1$*;*

(R2) $\mathbf{E}\left(f(\boldsymbol{y}, \hat{\boldsymbol{y}})\right)$ *is monotone decreasing in* p_k *if* $\hat{y}_k = 0$*.*

Furthermore, f *has a BOP of the form* (56)*.*

Thus, a BOP of the accuracy measure f can be constructed following a procedure similar to the case of f_β (Chai, 2005; Jansche, 2007; Ye et al., 2012). First, the labels are sorted according to their marginal probabilities. Then, we evaluate all possible thresholds t and find the one maximizing the expected value of f. Obviously, the computation of this expectation essentially amounts to computing the expected value $\mathbf{E}\left(f(\boldsymbol{y}, \hat{\boldsymbol{y}})\right)$. If this can be accomplished in time $O(\phi(K))$, then the overall complexity is upper-bounded by $O(K\phi(K))$, because there are $O(K)$ possibilities for the threshold. Note that this result holds for a wide range of MLC losses and accuracy measures including those presented in the Table 2.

In the following, we show that if f is an MLC accuracy measure satisfying (D4) and (D5) given in Definition 4, then its generalization (51) is semi-uncertainty-aligned, i.e., F has a BOP with decision set of the form (27).

Lemma 3. *Let* π *be a permutation that sorts the labels* λ_k *in decreasing order of their marginal probabilities* $p_k = \boldsymbol{p}_k(1 \mid \boldsymbol{x})$*. Let* f *be an MLC accuracy measure of the form* (11)*, which is monotonic in the sense of Definition 4. Furthermore, assume CLI in the sense of* (15)*. Then, for any* $d \in \{0, 1, \dots, K\}$*, the solution* $\hat{\boldsymbol{y}}^d$ *of the inner maximization* (52) *is a decision set of the form* $\langle\!\langle l, r \rangle\!\rangle$ *with* $d = K - r + l + 1$*.*

Proposition 5. *Given the assumptions of Lemma 3, the generalization (51) of* f *has a BOP* $\hat{\boldsymbol{y}}$ *with decision set of the form* $D(\hat{\boldsymbol{y}}) = \langle\!\langle l, r \rangle\!\rangle$*.*

Thanks to this result, a BOP of the generalized accuracy measure F can be constructed following a procedure similar to the case of the rank loss. First, the labels are sorted according to their marginal probabilities. Then, we evaluate all possible partial predictions $\hat{\boldsymbol{y}}$ with decision sets of the form $\langle\!\langle l, r \rangle\!\rangle$, and find the one for which the expected value of F is maximal. Obviously, the computation of this expectation essentially amounts to computing the expected value $\mathbf{E}\left(f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right)$. If this can be accomplished in time $O(\phi(K))$, then the overall complexity is upper-bounded by $O(K^2\phi(K))$, because there are $O(K^2)$ possibilities for the decision set.

7.3 BOP for the F-Measure and Jaccard Measure

In the following, we show that, by exploiting specific properties of MLC accuracy measures, this complexity can even be reduced further. A representative example is the F_β measure, for which the computation of $\mathbf{E}\left(f_\beta(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right)$ requires time $O(K^2)$ (Decubber et al., 2018; Ye et al., 2012). However, for finding the BOP of F_β, an algorithm of complexity $O(K^3)$ instead of $O(K^4)$ can be devised, essentially by (re-)computing the expectations $\mathbf{E}\left(f_\beta(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right)$ in a clever way using dynamic programming (Decubber et al., 2018; Waegeman et al., 2014). A similar result can be shown for the Jaccard measure.

Lemma 4. *Given a query instance* $\boldsymbol{x}$*, assume marginal probabilities* p_k*,* $k \in [K]$*, are made available by an MLC predictor* $\mathbf{h}$*. Let* π *be the permutation that sorts the labels in decreasing order of these* *probabilities. Denote by*

$$Q(l, l_1) := \boldsymbol{p}\left(\sum_{k=1}^{l} y_{\pi(k)} = l_1 \,\middle|\, \boldsymbol{x} \right) , \tag{57}$$

$$P(r', r'_1) := \boldsymbol{p}\left(\sum_{k=r}^{K} y_{\pi(k)} = r'_1 \,\middle|\, \boldsymbol{x} \right) , \tag{58}$$

where $r' := K + 1 - r$. Thus, $Q(l, l_1)$ is the probability to find l_1 positives among the first l labels, and $P(r', r'_1)$ the probability to have r'_1 positives among the last r ones. Assuming CLI in the sense of (15)*, the quantities $Q(l, l_1)$, $0 \leq l_1 \leq l \leq K$, and $P(r', r'_1)$, $0 \leq r'_1 \leq r' \leq K$, can be determined in time $O(K^2)$.*

The quantities $Q(l, l_1)$ and $P(r', r'_1)$ will be used when computing BOPs of F_β and F_{Jac} in the following propositions.

Proposition 6. *Suppose the assumptions of Lemma 4 to hold. A BOP of the generalized accuracy measure F_β can be found in time $O(K^3)$ using Algorithm 2.*

Algorithm 2 Determining a BOP of the generalized measure F_β

```
Input: marginal probabilities $\boldsymbol{p} = (p_1, p_2, \ldots, p_K)$, penalty $g(\cdot)$, $\beta$
$\boldsymbol{p} \longleftarrow$ sort($\boldsymbol{p}$) s.t. $p_1 \geq p_2 \geq \ldots \geq p_K$
compute $Q \longleftarrow$ using Lemma 4
$F_\beta(0, K+1) \longleftarrow 1 - g(K)$
$l_0 \longleftarrow 0$, $r_0 \longleftarrow K + 1$
for $l = 1$ to $K$ do
    for $i = 0$ to $K$ do
        initialize $S(l, i) \longleftarrow \frac{1}{l\beta^{-2} + i}$
    end for
    $F_\beta(l, K+1) \longleftarrow \beta' \sum_{l_1=0}^{l} l_1 Q(l, l_1) S(l, l_1) - g(K - l)$
    for $r = K$ to $l + 1$ do
        for $i = 0$ to $K - l - r'$ do
            $S(l, i) \longleftarrow p_r S(l, i+1) + \left(1 - p_r\right) S(l, i)$
        end for
        $F_\beta(l, r) \longleftarrow \beta' \sum_{l_1=0}^{l} l_1 Q(l, l_1) S(l, l_1) - g(r - l - 1)$
    end for
    $r_l \longleftarrow \text{argmax}_r\, F_\beta(l, r)$ , $F_\beta(\hat{\boldsymbol{y}}^l) \longleftarrow F_\beta(l, r_l)$
end for
$l^* \longleftarrow \text{argmax}_l\, F_\beta(\hat{\boldsymbol{y}}^l)$
Output: a BOP $\hat{\boldsymbol{y}} := \hat{\boldsymbol{y}}^{l^*}_{r_{l^*}}$
```

Another well-known measure in MLC is the Jaccard measure

$$f_{\text{Jac}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{tp}{tp + fn + fp} = \frac{\sum_{k=1}^{K} y_k \hat{y}_k}{\sum_{k=1}^{K} y_k + \sum_{k=1}^{K} \hat{y}_k - \sum_{k=1}^{K} y_k \hat{y}_k} . \tag{59}$$

Under the assumption of label independence, a BOP for f_{Jac} can be found in time $O(K^2)$ (Quevedo et al., 2012; Waegeman et al., 2014).

Proposition 7. *Under the assumption of CLI in the sense of* (15)*, a BOP for the generalized Jaccard measure F_{Jac} can be found in time $O(K^3)$ using Algorithm 3.*

Algorithm 3 Determining a BOP of the generalized measure F_{Jac}

```
Input: marginal probabilities $\boldsymbol{p} = (p_1, p_2, \dots, p_K)$, penalty $g(\cdot)$
$\boldsymbol{p} \longleftarrow$ sort($\boldsymbol{p}$) s.t. $p_1 \geq p_2 \geq \dots, \geq p_K$
compute $Q \longleftarrow$ using Lemma 4
$F_{\text{Jac}}(0, K+1) \longleftarrow 1 - g(K)$
$l_0 \longleftarrow 0, r_0 \longleftarrow K + 1$
for $l = 1$ to $K$ do
    for $i = l$ to $K$ do
        initialize $S(i) \longleftarrow \frac{1}{i}$
    end for
    $F_{\text{Jac}}(l, K+1) \longleftarrow S(l) \sum_{l_1=0}^{l} l_1 Q(l, l_1) - g(K - l)$
    for $r = K$ to $l + 1$ do
        for $i = l + r'$ to $K$ do
            $S(i) \longleftarrow p_r S(i) + \left(1 - p_r\right) S(i-1)$
        end for
        $F_{\text{Jac}}(l, r) \longleftarrow S(l + r') \sum_{l_1=0}^{l} l_1 Q(l, l_1) - g(r - l - 1)$
    end for
    $r_l \longleftarrow \text{argmax}_r \, F_{\text{Jac}}(l, r)$ , $F_{\text{Jac}}(\hat{\boldsymbol{y}}^l) \longleftarrow F_{\text{Jac}}(l, r_l)$
end for
$l^* \longleftarrow \text{argmax}_l \, F_{\text{Jac}}(\hat{\boldsymbol{y}}^l)$
Output: a BOP $\hat{\boldsymbol{y}} := \hat{\boldsymbol{y}}^{l^*}_{r_{l^*}}$
```

8. Related Work

In spite of extensive research on multilabel classification in the recent past, there is surprisingly little work on abstention in MLC. A notable exception is an approach by Pillai et al. (2013). The authors follow the principle of empirical utility maximization and focus on the f_β measure as a performance metric. More specifically, they tackle the problem of optimizing the f_β measure on a subset of label predictions, subject to the constraint that the effort for manually providing the remaining labels (those on which the learner abstains) does not exceed a pre-defined value f_{max}. The decision whether or not to abstain on a label is guided by two thresholds on the predicted degree of relevance, which are tuned in a suitable manner.

More precisely, Pillai et al. (2013) assume that MLC with partial abstention can be implemented in the form of a generalized thresholded scoring classifier, which means that

$$\hat{y}_k = h_k(\boldsymbol{x}; t_l^k, t_r^k) = \begin{cases} 1 & \text{if } s_k(\boldsymbol{x}) > t_r^k(\boldsymbol{x}) \\ \perp & \text{if } t_l^k(\boldsymbol{x}) \leq s_k(\boldsymbol{x}) \leq t_r^k(\boldsymbol{x}) \\ 0 & \text{if } s_k(\boldsymbol{x}) < t_l^k(\boldsymbol{x}) \end{cases} , \tag{60}$$

where $s_k(\cdot)$ is a (real-valued) scoring function provided by an MLC classifier and $t_l^k(\boldsymbol{x}) \leq t_r^k(\boldsymbol{x}) \in \mathbb{R}$ are thresholds. Associating each classifier $\boldsymbol{h}(\cdot;\boldsymbol{t})$ with its parameters

$$\boldsymbol{t} = \left(t_l^1, t_r^1, t_l^2, t_r^2, \ldots, t_l^K, t_r^K\right) ,$$

the problem of risk minimization comes down to finding the optimal thresholds $\boldsymbol{t}$ on the training data $\mathcal{D}$, i.e.,

$$\begin{aligned} \hat{\boldsymbol{t}} \in \operatorname*{argmax}_{\boldsymbol{t}\in\mathbb{R}^{2K}} & \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} f_\beta(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) \\ \text{s.t.} & \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} f(\boldsymbol{y}_A, \hat{\boldsymbol{y}}_D) \leq f_{max} , \end{aligned}$$

where $\hat{\boldsymbol{y}} = \boldsymbol{h}(\boldsymbol{x};\boldsymbol{t})$ and $f(\boldsymbol{y}_A, \hat{\boldsymbol{y}}_D)$ is the cost for manually providing the labels on $A(\boldsymbol{y})$.

More indirectly related is work on uncertainty in multilabel classification. In particular, Park and Simoff (2015) propose a modification of the entropy measure to quantify the uncertainty of an MLC prediction, and show that this measure correlates with the accuracy of the prediction. Hence, they conclude that it could be used as a measure of acceptance (and hence rejection) of a prediction. While the focus here is on the uncertainty of a complete labeling $\boldsymbol{y}$, Destercke (2015) and Antonucci and Corani (2017) quantify the uncertainty in individual predictions y_k using imprecise probabilities and so-called credal classifiers, respectively. Again, corresponding estimates can be used for the purpose of producing more informed decisions, including partial predictions.

In most of the cases analyzed in this paper, an optimal prediction policy is to sort the labels according to their (predicted) probability of relevance, and to abstain on those labels in the middle, i.e., to predict only on those labels exceeding a certain upper threshold or remaining below a lower threshold on the probability. Therefore, our methods are in a sense also related to methods for threshold optimization in standard MLC (although our thresholds are not fixed but defined implicitly through the optimal size of the abstention set). Indeed, even in standard MLC, the optimal probability threshold for predicting positive is not necessarily $1/2$. Instead, tuning the threshold separating between positive and negative predictions, perhaps even in a label-wise manner, may improve accuracy (Fan and Lin, 2007; Jasinska et al., 2016).

9. Experiments

In this section, we present an empirical analysis that is meant to show the effectiveness of our approach to prediction with abstention. To this end, we perform experiments on a set of standard benchmark data sets from the MULAN repository[2] (cf. Table 3), following a standard 10-fold cross-validation procedure. Binary relevance (BR) learning and ensembles of classifier chains (ECC) are employed as the MLC classifiers on these data sets. In addition, we perform experiments on an image data set, on which we train ensembles of convolutional neural networks. The data set consists of $2,000$ natural scene images, where the class labels are desert, mountains, sea, sunset and trees[3]. Because training deep networks is much more complex, we conduct a 3-fold (instead of a 10-fold) cross-validation procedure.

[2] http://mulan.sourceforge.net/datasets.html

[3] `https://www.lamda.nju.edu.cn/data_MIMLimage.ashx?AspxAutoDetectCookieSupport=1`.

#	NAME	# INST.	# FEAT.	# LAB.
1	CAL500	502	68	174
2	EMOTIONS	593	72	6
3	SCENE	2407	294	6
4	YEAST	2417	103	14
5	MEDIAMILL	43907	120	101
6	NUS-WIDE	269648	128	81

Table 3: Data sets used in the experiments

9.1 Experimental Setting

In the following, we describe the MLC classifiers used in experiments and the comparison criteria.

9.1.1 BR

Given our assumption of label independence, simple BR learning is in principle well justified for training MLC classifiers.[4] Besides, please note that we are first of all interested in analyzing the effectiveness of abstention, and less in maximizing overall performance. Indeed, all competitors essentially only differ in how the conditional probabilities provided by the learner are turned into a (partial) MLC prediction. Note that BR can be instantiated with different base learners. We perform experiments with two variants of BR, namely with logistic regression (in its default setting in sklearn, i.e., with regularisation parameter set to 1) as the base learner (BR+LR) and with support vector machines, using Platt-scaling (Lin et al., 2007; Platt, 1999) to turn scores into probabilities, as a base learner (BR+SVM).

9.1.2 ECC

As a state-of-the-art MLC method that is highly competitive in terms of predictive performance and able to take label dependencies into account, we additionally include ECC[5] (Read et al., 2011, 2021). Following the suggestion by Read et al. (2011), we learn MLC classifiers $\boldsymbol{h}^1, \ldots, \boldsymbol{h}^M$ via classifier chains over a (randomly chosen) set of M permutations of the labels. More specifically, each classifier chain $\boldsymbol{h}^m$ produces predictions in the form of scores in $[0, 1]^K$, which can be seen as *dependent* marginal probabilities, i.e., marginal probability estimates which to some extent take label dependence into account. For each λ_k, the final marginal probability produced by the ECC is then obtained by the arithmetic mean

$$\bar{p}_k = \frac{1}{M} \sum_{m=1}^{M} p_{k,m} \,, \tag{61}$$

where $p_{k,m}$ is the score produced by the ensemble member $\boldsymbol{h}^m$. Similar to the case of BR, we perform experiments with two variants of ECC, again using logistic regression (ECC+LR) and support vector machines (ECC+SVM) as base learners. The cardinality M of the ECCs is set to 50. For further technical details, we refer to (Nguyen et al., 2020).

[4] For an implementation in Python, see `http://scikit.ml/api/skmultilearn.base.problem_transformation.html`.

[5] For an implementation in Python, see `http://scikit.ml/api/skmultilearn.problem_transform.cc.html`.

9.1.3 CONVOLUTIONAL NEURAL NETWORKS

Ensembles of 5 VGG16-based convolutional neural network classifiers (EVGG16) [6] are employed in the experiments on the image data set. The network consists of a VGG16-based encoder (pretrained on ImageNet) whose layers are freezed, except the last convolutional block. Moreover, we add a fully connected classification layer head including a 128-neuron hidden dense layer and the 5-neuron classification head. The networks are trained using SGD with Nesterov momentum (lr $= 10^{-6}$, momentum $= 0.9$) for 100 epochs. The loss is binary cross-entropy for multi-label classification. The input images are resized to $128 \times 128 \times 3$. Detailed model summary with parameter sizes are given in Appendix F. Similar to the case of ECC, each classifier produces predictions in the form of scores in $[0, 1]^K$, which can be seen as marginal probabilities. For each λ_k, the final marginal probability produced by the EVGG16 is then obtained by the arithmetic mean (61).

9.1.4 COMPARISON CRITERIA

We compare the performance of reliable classifiers, which are allowed to abstain in cases of uncertainty, to the conventional classifier that makes full predictions (MLC) as well as the cost of full abstention (ABS), i.e., the classifier that always abstains on all labels — these two serve as baselines that MLC with abstention should be able to improve on. A reliable classifier is obtained as a risk-minimizer of the extension (29) of the MLC loss, instantiated by the penalty function g and the constant c. Two such instantiations are considered for (30) and (31):

- SEP with linear penalty $g_1(a) = a \cdot c$, and
- PAR with concave penalty $g_2(a) = (a \cdot K \cdot c')/(K + a)$.

The performance of a classifier is evaluated in terms of the average loss. Besides, we compute the average abstention size $|A(\hat{\boldsymbol{y}})|/K$.

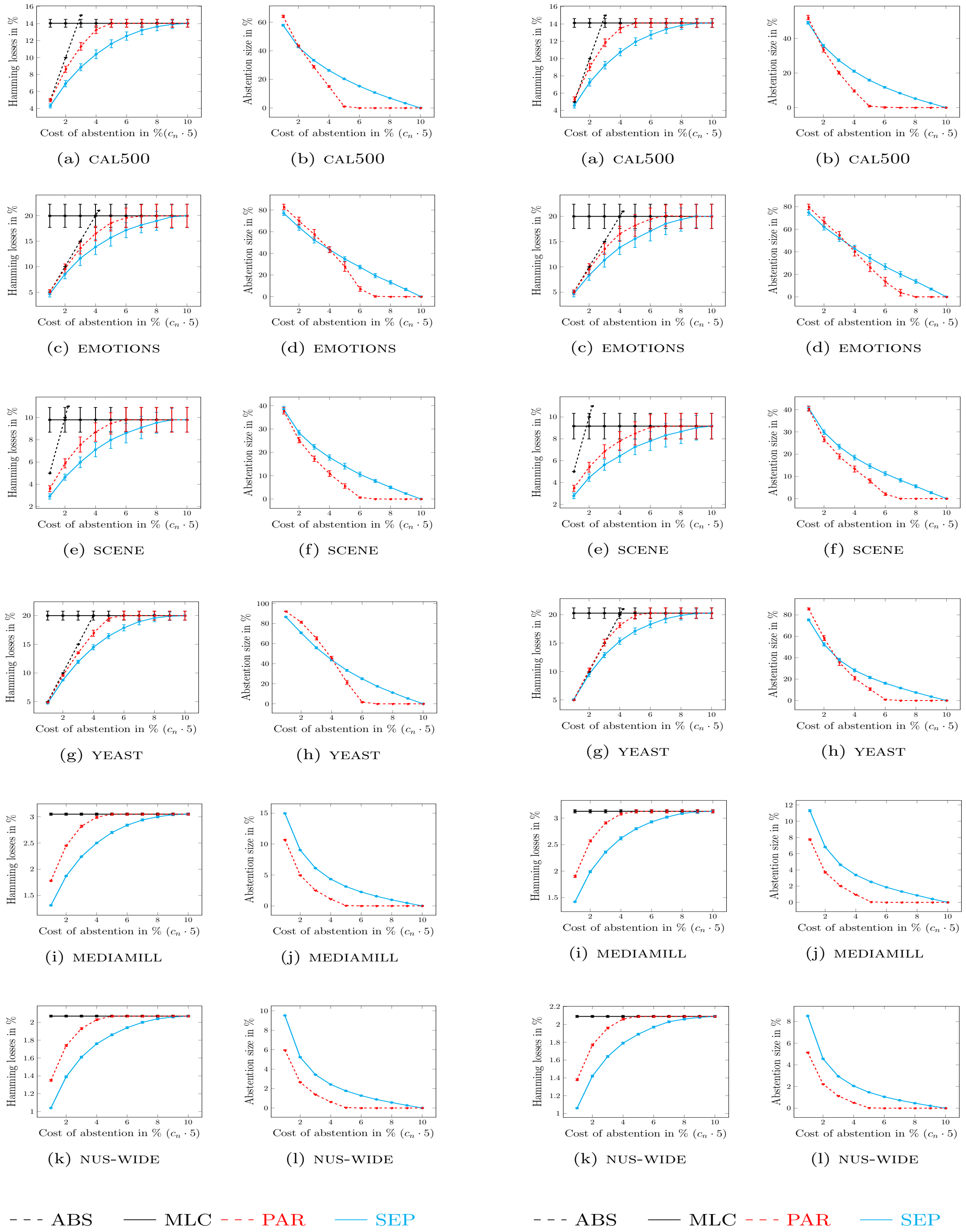
We conduct experiments for five MLC losses: Hamming loss, rank loss, subset 0/1 loss, F_1-measure, and Jaccard measure. Note that with the same cost of abstention c, the cost of making full abstention given by g_1 is twice the one given by g_2. To better visualize the effectiveness of partial abstention (compared to full abstention), the cost of abstention (horizontal axis) is chosen according to the cost of full abstention. Thus, the actual c' is twice the one given in the figures. For Hamming loss, $c \in [0.05, 0.5]$ and $c' \in [0.1, 1]$. In the case of the rank loss, $c \in [0.1, 1]$ and $c' \in [0.2, 2]$, and for subset 0/1 loss, $c \in [0.25/K, 2.5/K]$ and $c' \in [0.5/K, 5/K]$. Finally, $c \in [0.1/K, 1/K]$ and $c' \in [0.2/K, 2/K]$ for the F_1-measure and Jaccard measure. The cost of doing abstention varies mainly due to the range of the MLC loss. For example, the cost in the case of the F_1-measure should be smaller than the one of Hamming loss, since the F_1-measure (9) takes values in [0, 1], while the Hamming loss (7) takes values in $[0, K]$.

9.2 Results

In the following, we summarize the results for the case of BR+LR and ECC+LR. Similar results for BR+SVM and ECC+SVM are given in Appendix E.

- The results illustrated in Figure 1 clearly confirm our expectations. The Hamming loss under partial abstention is often much lower than the loss under full prediction and full abstention,

[6] For an implemetation in Python, see `https://github.com/julilien/MLCAmazonFromSpace`.



(a) CAL500 (b) CAL500 (a) CAL500 (b) CAL500

(c) EMOTIONS (d) EMOTIONS (c) EMOTIONS (d) EMOTIONS

(e) SCENE (f) SCENE (e) SCENE (f) SCENE

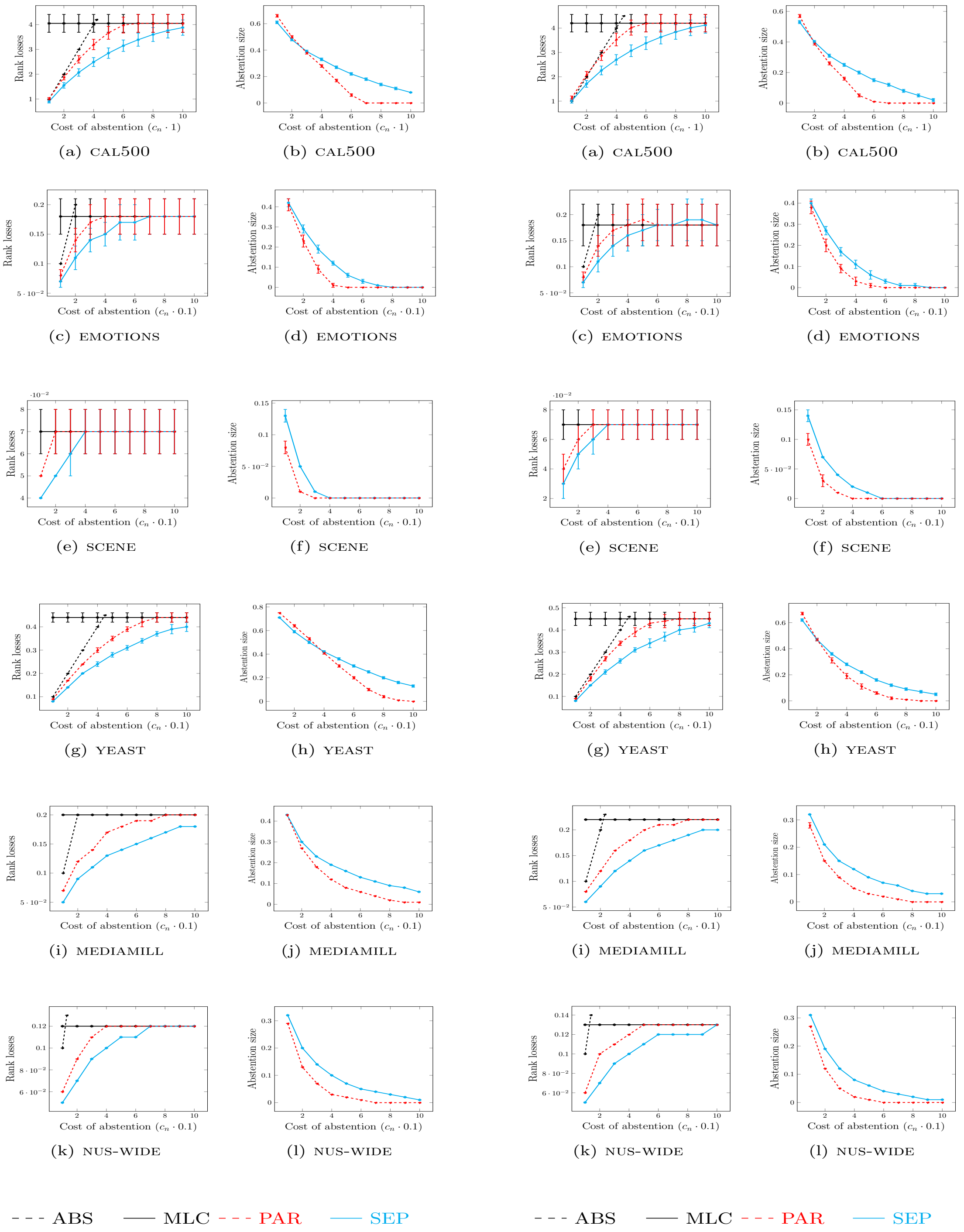
(g) YEAST (h) YEAST (g) YEAST (h) YEAST

(i) MEDIAMILL (j) MEDIAMILL (i) MEDIAMILL (j) MEDIAMILL

(k) NUS-WIDE (l) NUS-WIDE (k) NUS-WIDE (l) NUS-WIDE

a. BR+LR b. ECC+LR

Figure 1: Experimental results in terms of average Hamming loss (in percent), which is plotted in percent of the maximal loss K, and abstention size (in percent) for $g_1(a) = a \cdot c$ (SEP) and $g_2(a) = (a \cdot K \cdot c)/(K + a)$ (PAR), as a function of the cost of abstention.



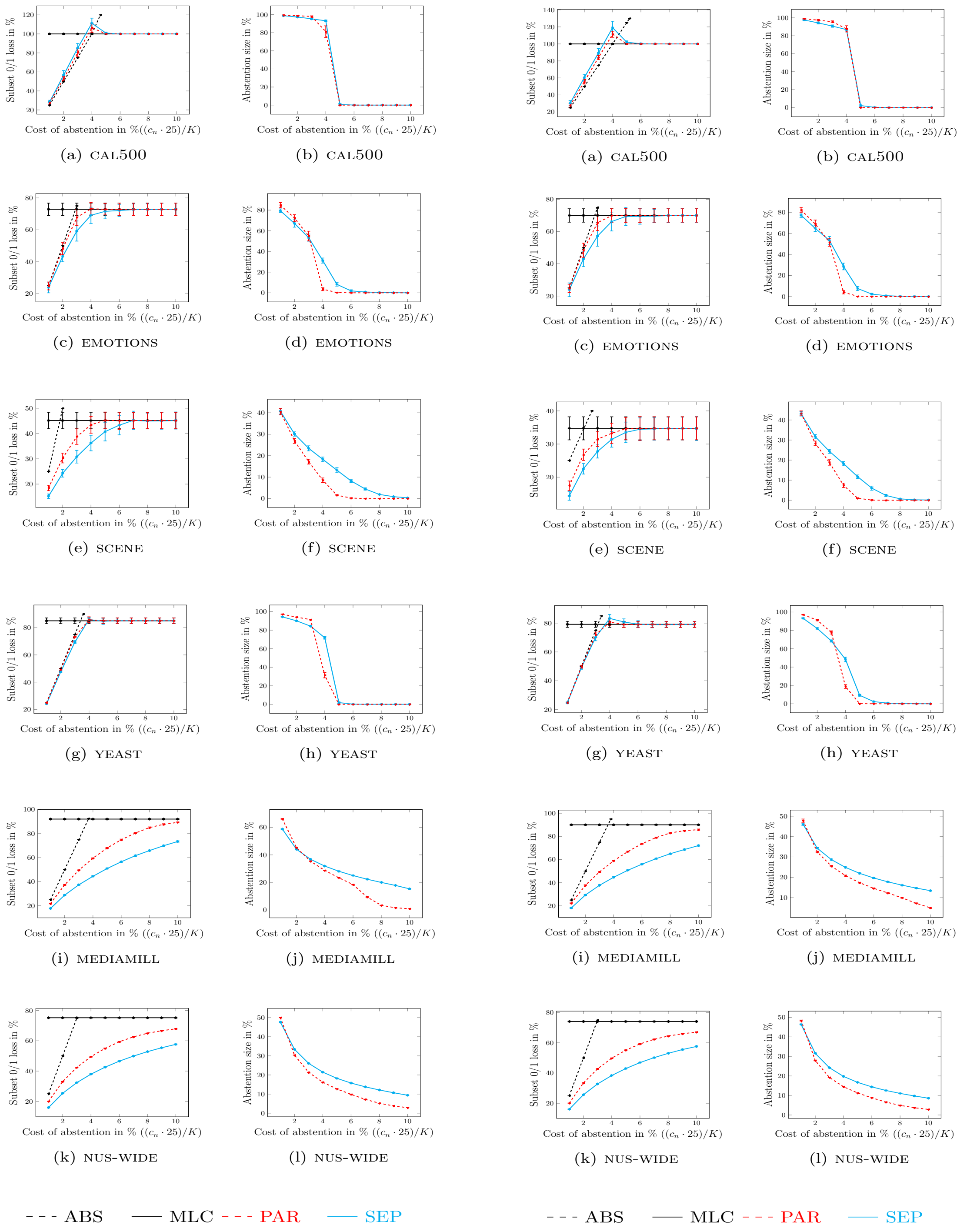
(a) CAL500 (b) CAL500 (c) EMOTIONS (d) EMOTIONS (e) SCENE (f) SCENE (g) YEAST (h) YEAST (i) MEDIAMILL (j) MEDIAMILL (k) NUS-WIDE (l) NUS-WIDE

a. BR+LR

(a) CAL500 (b) CAL500 (c) EMOTIONS (d) EMOTIONS (e) SCENE (f) SCENE (g) YEAST (h) YEAST (i) MEDIAMILL (j) MEDIAMILL (k) NUS-WIDE (l) NUS-WIDE

b. ECC+LR

Figure 2: Experimental results in terms of average rank loss L_R and abstention size for $g_1(a) = a \cdot c$ (SEP) and $g_2(a) = (a \cdot K \cdot c)/(K + a)$ (PAR), as a function of the cost of abstention.



(a) CAL500 (b) CAL500 (a) CAL500 (b) CAL500

(c) EMOTIONS (d) EMOTIONS (c) EMOTIONS (d) EMOTIONS

(e) SCENE (f) SCENE (e) SCENE (f) SCENE

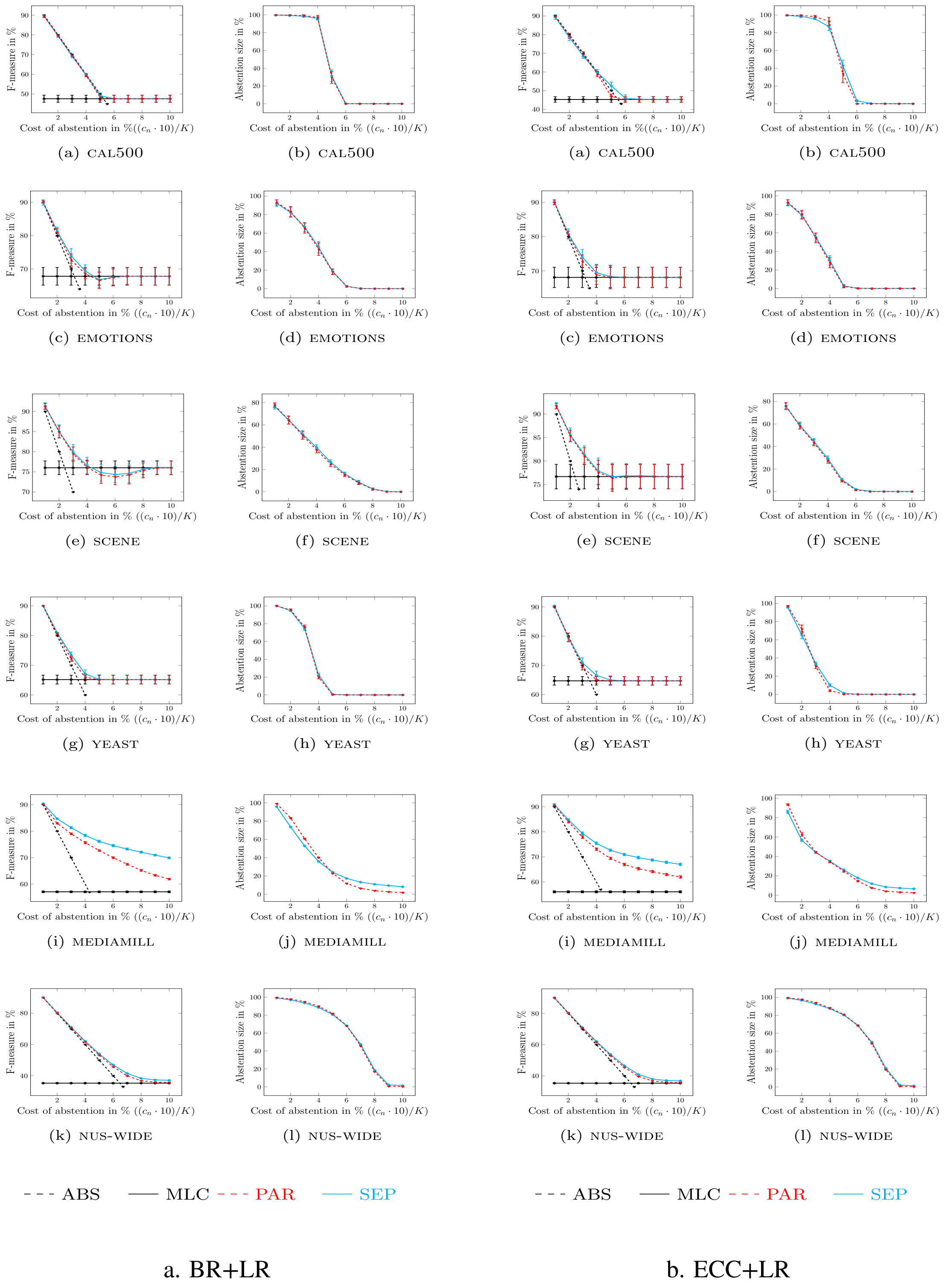
(g) YEAST (h) YEAST (g) YEAST (h) YEAST

(i) MEDIAMILL (j) MEDIAMILL (i) MEDIAMILL (j) MEDIAMILL

(k) NUS-WIDE (l) NUS-WIDE (k) NUS-WIDE (l) NUS-WIDE

a. BR+LR b. ECC+LR

Figure 3: Experimental results in terms of average Subset 0/1 loss L_S (in percent) and abstention size (in percent) for $g_1(a) = a \cdot c$ (SEP) and $g_2(a) = (a \cdot K \cdot c)/(K + a)$ (PAR), as a function of the cost of abstention.



(a) CAL500 (b) CAL500 (a) CAL500 (b) CAL500

(c) EMOTIONS (d) EMOTIONS (c) EMOTIONS (d) EMOTIONS

(e) SCENE (f) SCENE (e) SCENE (f) SCENE

(g) YEAST (h) YEAST (g) YEAST (h) YEAST

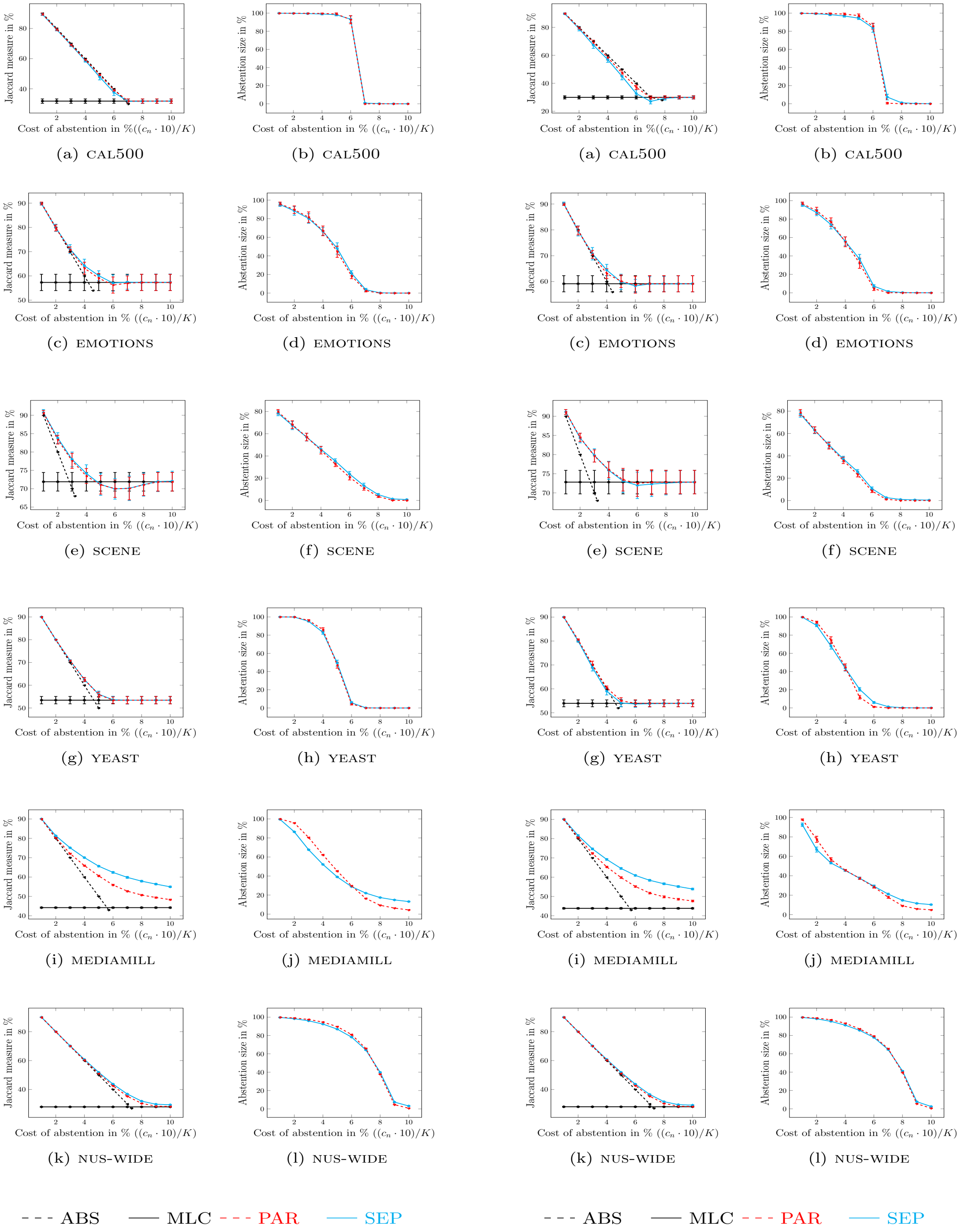
(i) MEDIAMILL (j) MEDIAMILL (i) MEDIAMILL (j) MEDIAMILL

(k) NUS-WIDE (l) NUS-WIDE (k) NUS-WIDE (l) NUS-WIDE

a. BR+LR

b. ECC+LR

Figure 4: Experimental results in terms of average F_1 (in percent) and abstention size (in percent) for $g_1(a) = a \cdot c$ (SEP) and $g_2(a) = (a \cdot K \cdot c)/(K + a)$ (PAR), as a function of the cost of abstention.



(a) CAL500 (b) CAL500 (a) CAL500 (b) CAL500

(c) EMOTIONS (d) EMOTIONS (c) EMOTIONS (d) EMOTIONS

(e) SCENE (f) SCENE (e) SCENE (f) SCENE

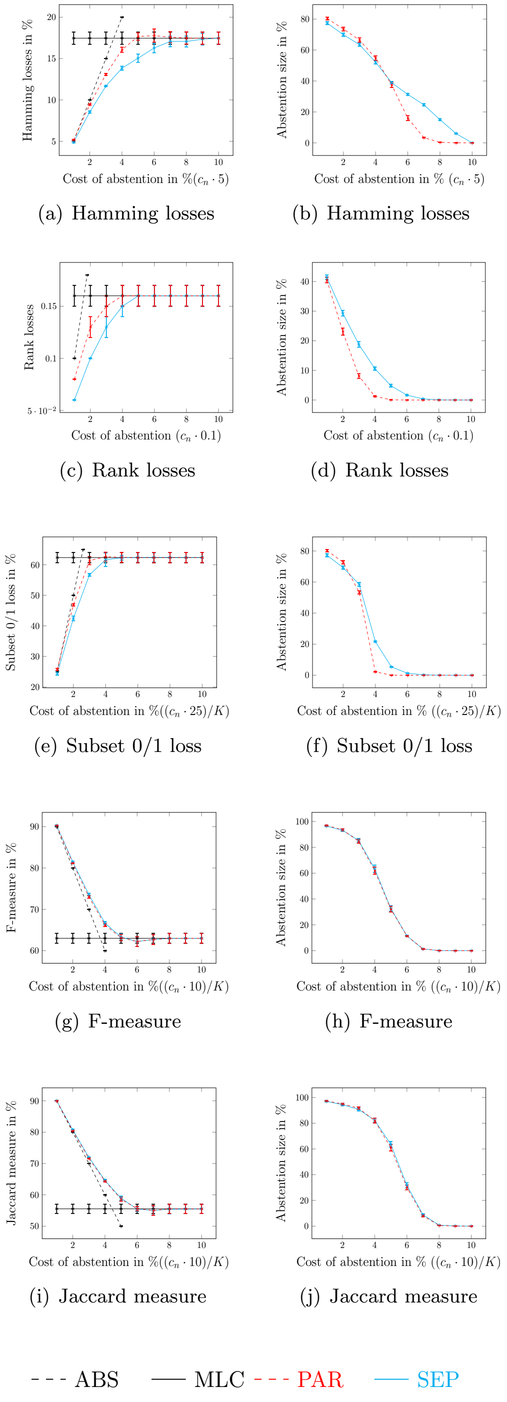
(g) YEAST (h) YEAST (g) YEAST (h) YEAST

(i) MEDIAMILL (j) MEDIAMILL (i) MEDIAMILL (j) MEDIAMILL

(k) NUS-WIDE (l) NUS-WIDE (k) NUS-WIDE (l) NUS-WIDE

a. BR+LR b. ECC+LR

Figure 5: Experimental results in terms of average Jaccard measure F_{Jac} (in percent) and abstention size (in percent) for $g_1(a) = a \cdot c$ (SEP) and $g_2(a) = (a \cdot K \cdot c)/(K + a)$ (PAR), as a function of the cost of abstention.



(a) Hamming losses

(b) Hamming losses

(c) Rank losses

(d) Rank losses

(e) Subset 0/1 loss

(f) Subset 0/1 loss

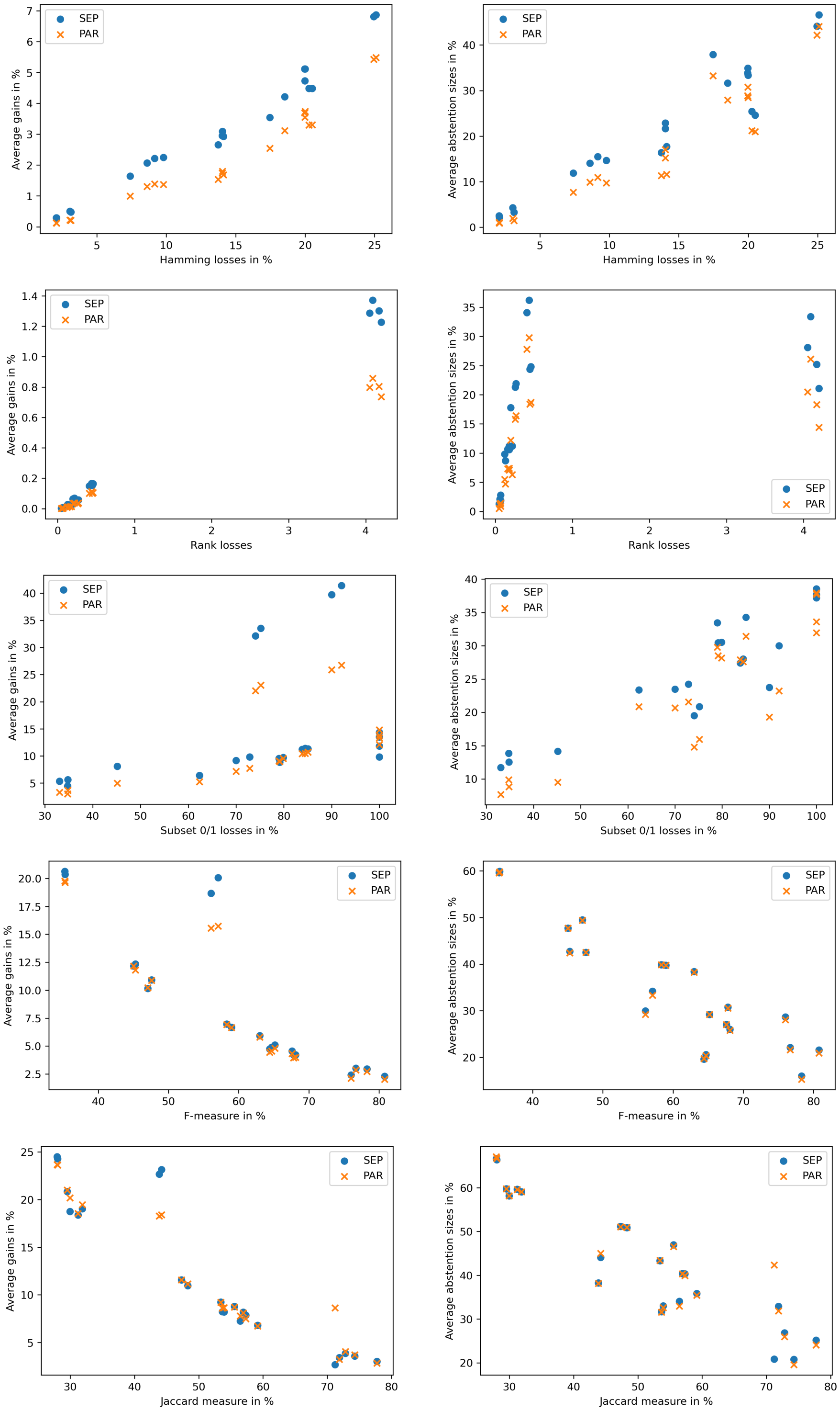
(g) F-measure

(h) F-measure

(i) Jaccard measure

(j) Jaccard measure

Figure 6: Experimental results for EVGG16 on the natural scene image data set.



a. Average gain

b. Average abstention size

Figure 7: Experimental results in terms of average gains and average abstention sizes (in percent) as functions of the predictive score of the conventional classifiers.

showing the effectiveness of the approach. When the cost c increases, the loss increases while the abstention size decreases, with a convergence of the performance of SEP and PAR to the one of MLC at $c = 0.5$ and $c' = 1$, respectively.

- Similar results are obtained in the case of rank loss (cf. Figure 2), except that convergence to the performance of MLC is slower (i.e., requires lager cost values c and c', especially on the data set CAL500). This is plausible, because the cost of a wrong prediction on a single label can be as high as $K - 1$, compared to only 1 in the case of Hamming loss.
- Results for the subset 0/1 Loss, F_1-measure, and Jaccard measure are illustrated in Figure 3, 4, and 5, respectively. Again, the results are very similar to those presented above.
- Also quite similar are the results obtained for EVGG16 on the natural scene image data set (cf. Figure 6).

Ideally, a reliable classifier should be more cautious on difficult data sets, on which the conventional classifier is likely to fail (Nguyen et al., 2018; Yang et al., 2014). In such cases, one should expect to observe a stronger tendency to abstain. We conducted a meta-analysis on the experimental results to verify this ability of our methods. For each configuration, i.e., a combination of data set, MLC loss, and MLC classifier, we average the gains (defined by the difference between the loss of the reliable classifier and the conventional classifier) and the abstention sizes over the tests for the different costs of abstention. For each MLC loss, we have 21 configurations with various levels of loss for the conventional classifiers in total. When considering the average gains and average abstention sizes as functions of the loss (accuracy) of the conventional classifier, we expect to see an increasing (decreasing) trend. The results illustrated in Figure 7 clearly confirm our expectation for all five MLC losses.

10. Conclusion

This paper presents a formal framework of MLC with partial abstention, which builds on two main building blocks: First, the extension of an underlying MLC loss function so as to accommodate abstention in a proper way, and second, the problem of optimal prediction, that is, minimizing this loss in expectation.

We instantiated our framework for several MLC losses which are important and commonly used loss functions in multilabel classification, including the Hamming loss, the rank loss, the subset 0/1 loss, and a family of confusion matrix-derived accuracy measures. We elaborated on properties of risk-minimizers, showed them to have a specific structure, and devised efficient methods to produce optimal predictions. Experimentally, we showed these methods to be effective in the sense of reducing loss when being allowed to abstain.

While we showed that a BOP of any decomposable loss can be found efficiently, regardless of whether the labels are (conditionally) independent or not, we have to make the assumption of conditional label independence when working with non-decomposable losses. An obvious direction for future work is to extend our formal framework toward non-decomposable losses under the assumption of possible label dependencies.

Acknowledgments

We thank Willem Waegeman and Julian Lienen for interesting discussions and useful suggestions. This work was supported by the Germany Research Foundation (DFG) under grant number 400845550.

Appendix A. Proofs for Section 4 (Decomposable Losses)

Proof of Proposition 1. Let π be the permutation sorts the labels in increasing order of the label-wise expected losses, i.e., $s_{\pi(1)} \leq \cdots \leq s_{\pi(K)}$, where

$$\begin{aligned} s_k &= \min_{\bar{y}_k \in \{0,1\}} \mathbf{E}(\ell_k(y_k, \bar{y}_k)) = \min_{\bar{y}_k \in \{0,1\}} \sum_{\boldsymbol{y} \in \mathcal{Y}} \ell_k(y_k, \bar{y}_k) \cdot \boldsymbol{p}(\boldsymbol{y} \,|\, \boldsymbol{x}) \\ &= \min_{\bar{y}_k \in \{0,1\}} \sum_{\boldsymbol{y} \in \mathcal{Y}} \ell_k(1 - \bar{y}_k, \bar{y}_k)(p_k)^{1-\bar{y}_k}(1-p_k)^{\bar{y}_k} \\ &= \min\left((1-p_k)\ell_k(0,1), p_k \ell_k(1,0)\right) . \end{aligned}$$

The problem of finding BOP of the generalized loss (29) can be expressed as

$$\hat{\boldsymbol{y}} = \operatorname*{argmin}_{\bar{\boldsymbol{y}} \in \mathcal{Y}^*} \mathbf{E}\left(L(\boldsymbol{y}, \bar{\boldsymbol{y}})\right) = \operatorname*{argmin}_{0 \leq d \leq K} \mathbf{E}\left(\ell(\boldsymbol{y}_D, \hat{\boldsymbol{y}}^d_D)\right) + g(K-d) .$$

It is easy to check that, $\forall\, d = 0, 1, \dots, K$, $\hat{\boldsymbol{y}}^d$ is specified by the index set

$$D(\hat{\boldsymbol{y}}^d) := \{k \in [K] \,|\, \pi^{-1}(k) \leq d\} \text{ , and, } \hat{y}^d_k = \operatorname*{argmin}_{\bar{y}_k \in \{0,1\}} \mathbf{E}(\ell_k(y_k, \bar{y}_k)) \text{ , } \forall k \in D(\hat{\boldsymbol{y}}^d)$$

since, $\forall\, \bar{\boldsymbol{y}} \in \mathcal{Y}^*_d$, we have constant penalty $g(K-d)$ and

$$\begin{aligned} \mathbf{E}\left(\ell(\boldsymbol{y}_D, \bar{\boldsymbol{y}}_D)\right) &= \sum_{k \in D(\bar{\boldsymbol{y}})} \left(\sum_{\boldsymbol{y} \in \mathcal{Y}} \ell_k(y_k, \bar{y}_k) \cdot \boldsymbol{p}(\boldsymbol{y} \,|\, \boldsymbol{x}) \right) = \sum_{k \in D(\bar{\boldsymbol{y}})} \ell_k(1 - \bar{y}_k, \bar{y}_k)(p_k)^{1-\bar{y}_k}(1-p_k)^{\bar{y}_k} \\ &\geq \sum_{k \in D(\bar{\boldsymbol{y}})} \min_{\bar{y}_k \in \{0,1\}} \ell_k(1 - \bar{y}_k, \bar{y}_k)(p_k)^{1-\bar{y}_k}(1-p_k)^{\bar{y}_k} = \sum_{k \in D(\bar{\boldsymbol{y}})} s_k \\ &\geq \sum_{i=1}^{d} s_{\pi(i)} = \sum_{k \in [K] \,|\, \pi^{-1}(k) \leq d} s_k = \mathbf{E}\left(\ell(\boldsymbol{y}_D, \hat{\boldsymbol{y}}^d_D)\right) . \end{aligned}$$

The second inequality holds because replacing any $i \leq d$ by $i' > d$ cannot decrease $s_{\pi(i)}$. □

Proof of Corollary 1. The proof is obvious because in the case of Hamming loss, the degrees of uncertainty

$$u_k = 2\min(p_k, 1-p_k) = 2 \min_{\bar{y}_k \in \{0,1\}} \mathbf{E}(\ell_k(y_k, \bar{y}_k)) = 2 s_k .$$

Thus, sorting the labels in increasing order of the degrees of uncertainty u_k (24) is equivalent to doing so with the label-wise expected losses s_k. To this end, the proof of Corollary 1 is carried out consequently from the proof of Proposition 1. □

Proof of Corollary 2. It is easy to verify that the extension (29) of the Hamming loss is uncertainty-aligned since its risk-minimizers are always of the form (39).

In the following, we show that the BOP of the generalized Hamming loss (30) can be found simply by abstaining those labels with $\min(p_k, 1-p_k) > c$ and the predictions on the remainder $D(\hat{\boldsymbol{y}})$ are

$$\hat{y}_k = \operatorname*{argmin}_{\bar{y}_k \in \{0,1\}} (p_k)^{1-\bar{y}_k}(1-p_k)^{\bar{y}_k} . \tag{62}$$

The expected loss of the generalized Hamming loss (30) associated to $\bar{\boldsymbol{y}} \in \mathcal{Y}^*$ is

$$\begin{aligned}\mathbf{E}\left(L_H(\boldsymbol{y},\bar{\boldsymbol{y}})\right) &= \sum_{\boldsymbol{y}\in\mathcal{Y}} L_H(\boldsymbol{y},\bar{\boldsymbol{y}})\cdot \boldsymbol{p}(\boldsymbol{y}\mid\boldsymbol{x}) = \sum_{\substack{k:\bar{y}_k=0\\ k\in D(\bar{\boldsymbol{y}})}} p_k + \sum_{\substack{k:\bar{y}_k=1\\ k\in D(\bar{\boldsymbol{y}})}} (1-p_i) + |A(\bar{\boldsymbol{y}})|\cdot c\\ &= \sum_{\substack{k:\bar{y}_k=0\\ k\in D(\bar{\boldsymbol{y}})}} p_k + \sum_{\substack{k:\bar{y}_k=1\\ k\in D(\bar{\boldsymbol{y}})}} (1-p_k) + \sum_{k\in A(\bar{\boldsymbol{y}})} c .\end{aligned}$$

Finding the BOP is thus equivalent to solving the following optimization problem:

$$\hat{\boldsymbol{y}} = \operatorname*{argmin}_{\bar{\boldsymbol{y}}\in\mathcal{Y}^*} \mathbf{E}\left(L_H(\boldsymbol{y},\bar{\boldsymbol{y}})\right) = \operatorname*{argmin}_{\bar{\boldsymbol{y}}\in\mathcal{Y}^*} \left(\sum_{k:\bar{y}_k=0} p_k + \sum_{k:\bar{y}_k=1} (1-p_k) + \sum_{k:\bar{y}_k=\perp} c \right) .$$

Thus to minimize the expected loss, we should abstain all the index $k \in [K]$ s.t $c < \min(p_k, 1-p_k)$ and return an optimal d-prediction $D(\hat{\boldsymbol{y}}) := \{k | c > \min(p_k, 1-p_k)\}$. The proof of Corollary 2 is completed by predicting the labels on $D(\hat{\boldsymbol{y}})$ according to (62). □

Proof of Remark 1. For a seek of simplicity, let us denote by $a := |A(\hat{\boldsymbol{y}})|$ the number of abstained labels in $\hat{y}$. We start with the general setting that if

$$g(a) - g(a-1) \in [0,1],\ \forall a \in [K] , \tag{63}$$

the generalized Hamming loss (29) is monotonic.

Let us consider two predictions $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{y}}'$, s.t, for a given $k \in [K]$, we have

$$\begin{cases} \ell_H(y_k, \hat{y}_k) \prec \ell_H(y_k, \hat{y}'_k), & \text{and}\\ \ell_H(y_i, \hat{y}_i) = \ell_H(y_i, \hat{y}'_i), & \text{if } i \neq k, \end{cases}$$

where $\ell_H(y_k, \hat{y}_k)$ can be: $\ell_{\text{incorrect}}$ (am incorrect prediction), ℓ_{correct} (a correct prediction), and $\ell_{\text{abstention}}$ (an abstention). The preference relation $\prec$ is defined s.t, $\ell_{\text{incorrect}} \prec \ell_{\text{abstention}} \prec \ell_{\text{correct}}$.

We proceed by considering three possible combinations of the relation $\ell_H(y_k, \hat{y}_k) \prec \ell_H(y_k, \hat{y}'_k)$. The number of abstention in $\hat{\boldsymbol{y}}'$ can be either $a' \in \{a-1, a, a+1\}$.

- $\ell_{\text{incorrect}} \prec \ell_{\text{correct}}$: in this case, we have $a' = a$ and $D(\hat{\boldsymbol{y}}') = D(\hat{\boldsymbol{y}})$. It is clear that $L(\boldsymbol{y}, \hat{\boldsymbol{y}}) \geq L(\boldsymbol{y}, \hat{\boldsymbol{y}}')$ since

$$L_H(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{\substack{j\in D(\hat{\boldsymbol{y}})\\ i\neq k}} \ell_H(y_k, \hat{y}_k) + 1 + g(a) \geq \sum_{\substack{j\in D(\hat{\boldsymbol{y}}')\\ i\neq k}} \ell_H(y_i, \hat{y}_i) + g(a) = L_H(\boldsymbol{y}, \hat{\boldsymbol{y}}') .$$

- $\ell_{\text{incorrect}} \prec \ell_{\text{abstention}}$: in this case, we have $a' = a + 1$ and $D(\hat{\boldsymbol{y}}') = D(\hat{\boldsymbol{y}}) \setminus \{k\}$. We can easily validate that $L_H(\boldsymbol{y}, \hat{\boldsymbol{y}}) \geq L_H(\boldsymbol{y}, \hat{\boldsymbol{y}}')$ using the following analysis. Since $g(a+1) \leq 1 + g(a)$, thus

$$L_H(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{\substack{i \in D(\hat{\boldsymbol{y}}) \\ i \neq k}} \ell_H(y_i, \hat{y}_i) + 1 + g(a) \geq \sum_{i \in D(\hat{\boldsymbol{y}}')} \ell_H(y_i, \hat{y}_i) + g(a+1) = L_H(\boldsymbol{y}, \hat{\boldsymbol{y}}') \,.$$

- $\ell_{\text{abstention}} \prec \ell_{\text{correct}}$: in this case, we have $a' = a - 1$ and $D(\hat{\boldsymbol{y}}') \setminus \{i\} = D(\hat{\boldsymbol{y}})$. It is not difficult to see that $L_H(\boldsymbol{y}, \hat{\boldsymbol{y}}) \geq L_H(\boldsymbol{y}, \hat{\boldsymbol{y}}')$. Since $g(a) \geq g(a-1)$, thus

$$L_H(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{i \in D(\hat{\boldsymbol{y}})} \ell_H(y_i, \hat{y}_i) + g(a) \geq \sum_{\substack{j \in D(\hat{\boldsymbol{y}}') \\ i \neq k}} \ell_H(y_i, \hat{y}_i) + 0 + g(a-1) = L_H(\boldsymbol{y}, \hat{\boldsymbol{y}}') \,.$$

□

Appendix B. Proofs for Section 5 (Rank Loss)

Proof of Lemma 1. Let $D_d = \{k_1, \dots, k_d\}$ specify a partial prediction of size d, and let $\boldsymbol{y}_d$ be the labeling restricted to the selected labels. Since $c(\boldsymbol{y}) = s(\boldsymbol{y})(K - s(\boldsymbol{y}))$, then

$$\begin{aligned}
\mathbf{E}(c(\boldsymbol{y}_d)) &= \mathbf{E}\left(\left(\sum_{1 \leq i \leq d} y_{\pi(k_i)} \right) \left(d - \sum_{1 \leq i \leq d} y_{\pi(k_i)} \right) \right) \\
&= \mathbf{E}\left(d \left(\sum_{1 \leq i \leq d} y_{\pi(k_i)} \right) \right) - \mathbf{E}\left(\left(\sum_{1 \leq i \leq d} y_{\pi(k_i)} \right)^2 \right) \\
&= d \sum_{1 \leq i \leq d} \mathbf{E}(y_{\pi(k_i)}) - \sum_{1 \leq i,j \leq d} \mathbf{E}(y_{\pi(k_i)} y_{\pi(k_j)}) \\
&= (d-1) \sum_{1 \leq i \leq d} \mathbf{E}(y_{\pi(k_i)}) - \sum_{1 \leq i \neq j \leq d} \mathbf{E}(y_{\pi(k_i)} y_{\pi(k_j)}) \\
&= (d-1) \sum_{1 \leq i \leq d} p_{\pi(k_i)} - \sum_{1 \leq i \neq j \leq d} p_{\pi(k_i)} p_{\pi(k_j)} \,,
\end{aligned}$$

where we exploited that $(y_i)^2 = y_i$ and the assumption of (conditional) independence as made in the proposition.

According to (44) and (45), we can write the expected loss of a ranking π_{D_d}

$$\begin{aligned}
\mathbf{E}\left(\ell_R(\boldsymbol{y}, \pi_{D_d}) \right) &= \frac{1}{2} \mathbf{E}\left(\left(\ell_R(\boldsymbol{y}, \pi_{D_d}) - \ell_R(\boldsymbol{y}, \bar{\pi}_{D_d}) \right) \right) + \frac{1}{2} \mathbf{E}(c(\boldsymbol{y})) \\
&= \frac{1}{2} \left(\sum_{1 \leq i \leq d} (2i - (d+1)) p_{\pi(k_i)} + (d-1) \sum_{1 \leq i \leq d} p_{\pi(k_i)} - \sum_{1 \leq i \neq j \leq d} p_{\pi(k_i)} p_{\pi(k_j)} \right) \\
&= \sum_{1 \leq i \leq d} (i-1) p_{\pi(k_i)} - \sum_{1 \leq i < j \leq d} p_{\pi(k_i)} p_{\pi(k_j)} \\
&= \sum_{1 \leq i < j \leq d} p_{\pi(k_j)} (1 - p_{\pi(k_i)}) \,.
\end{aligned} \tag{64}$$

Next, we show that the expression (64) is minimized by a selection of the form (47), i.e.,

$$K_d = \langle\!\langle l, r \rangle\!\rangle = \{1, \ldots, l\} \cup \{r, \ldots, K\} ,$$

where $d = K - r + l + 1$, as stated in the lemma. To this end, note that the derivative of (64) with respect to u is given by

$$\delta_u = \sum_{i<u} (1 - p_{\pi(k_i)}) - \sum_{j>u} p_{\pi(k_j)} .$$

Thus, recalling that $p_{\pi(1)} \geq p_{\pi(2)} \geq \cdots \geq p_{\pi(K)}$, we can conclude that (64) can be reduced (or at least kept equal) if, for some $u \in \{1, \ldots, d\}$,

(i) $\delta_u \leq 0$ and $u - 1 \notin K_d$,

(ii) $\delta_u \geq 0$ and $u + 1 \notin K_d$,

namely by replacing u with $u - 1$ in D_d in case (i) and replacing u with $u + 1$ in case (ii). Let us call such a replacement a "swap".

Now, suppose that, contrary to the claim of the lemma, an optimal selection is not of the form (47) and cannot be improved by a swap either. Then we necessarily have a situation where $b_1, b_2, \ldots, b_u \in D_d$ is a block of consecutive indices such that $b_1 - 1 \notin D_d$ and $b_u + 1 \notin D_d$. Moreover, let l' be the largest index in D_d smaller than b_1 and r' the smallest index in D_d bigger than b_u. Since a swap from b_1 to $b_1 - 1$ is not valid,

$$\delta_{b_1} = \sum_{k \leq l'} (1 - p_{\pi(k_i)}) - \left(p_{\pi(b_2)} + \ldots + p_{\pi(b_u)} + \sum_{j \geq r'} p_{\pi(k_j)} \right) > 0 .$$

Likewise, since a swap from b_u to $b_u + 1$ is not valid,

$$-\delta_{b_u} = - \sum_{i \leq l'} (1 - p_{\pi(k_i)}) - \sum_{j=1}^{u-1} (1 - p_{\pi(b_j)}) + \sum_{j \geq r'} p_{\pi(k_j)} > 0 .$$

Summing up these two inequalities yields

$$p_{\pi(b_1)} - p_{\pi(b_u)} > u - 1 ,$$

which is a contradiction.

□

Proof of Lemma 2. We proceed under the assumption that $p_k \notin \{0, 1\}, \forall k \in [K]$. Let $D_d = \langle\!\langle l, r \rangle\!\rangle$ be an optimal d-selection (47) for $d \geq 2$. Since D_d is an optimal d-selection, neither a replacement from l to $r - 1$ nor a replacement from r to $l + 1$ on D_d reduces the expected loss. Denote by δ_l^d and δ_r^d the derivative of $\mathbf{E}\left(\ell_R(\boldsymbol{y}, \pi_{D_d})\right)$ with respect to l and r, thus,

$$\delta_l^d = \sum_{i \leq l-1} (1 - p_{\pi(i)}) - \sum_{r \leq j} p_{\pi(j)} \leq 0 , \tag{65}$$

$$\delta_r^d = \sum_{i \leq l} (1 - p_{\pi(i)}) - \sum_{r+1 \leq j} p_{\pi(j)} \geq 0 . \tag{66}$$

Lemma 1 implies that there is an optimal $(d+1)$-selection $\mathbf{K}_{d+1} = \langle\!\langle l', r' \rangle\!\rangle$. Denote by $\delta_{l'}$ and $\delta_{r'}$, the derivative of $\mathbf{E}\left(\ell_R(\boldsymbol{y}, \pi_{\mathbf{K}_{d+1}})\right)$ with respect to l' and r', thus

$$\delta_{l'}^{d+1} = \sum_{i \le l'-1} (1 - p_{\pi(i)}) - \sum_{j \ge r'} p_{\pi(j)} \le 0\,,$$
$$\delta_{r'}^{d+1} = \sum_{i \le l'} (1 - p_{\pi(i)}) - \sum_{j \ge r'+1} p_{\pi(j)} \ge 0\,.$$

Now, suppose that, contrary to the claim of the lemma,

$$\left(\langle\!\langle l', r' \rangle\!\rangle \neq \langle\!\langle l+1, r \rangle\!\rangle\right) \wedge \left(\langle\!\langle l', r' \rangle\!\rangle \neq \langle\!\langle l, r-1 \rangle\!\rangle\right).$$

Thus, $\mathbf{K}_{d+1}$ has 2 following possible forms: (i) $(l < l') \wedge (r < r')$ or (ii) $(l' < l) \wedge (r' < r)$.

The proof of Lemma 2 is completed by showing that both (i) and (ii) lead to the contradiction.

- $(l < l') \wedge (r < r')$: it is not difficult to see that

$$\left(\sum_{i \le l} (1 - p_{\pi(i)}) \le \sum_{i \le l'-1} (1 - p_{\pi(i)})\right) \wedge \left(-\sum_{r+1 \le j} p_{\pi(j)} \le -\sum_{r' \le j} p_{\pi(j)}\right).$$

  Furthermore, the equality can not occur in both inequalities at the same time, otherwise

$$(l = l' - 1) \wedge (r + 1 = r') \Rightarrow r - l = r' - l'\,.$$

  Thus, $\delta_r^d < \delta_{l'}^{d+1} \le 0$, that contradicts (66).

- $(l' < l) \wedge (r' < r)$: it is not difficult to see that

$$\left(\sum_{i \le l-1} (1 - p_{\pi(i)}) \ge \sum_{i \le l'} (1 - p_{\pi(i)})\right) \wedge \left(-\sum_{r \le j} p_{\pi(j)} \ge -\sum_{r'+1 \le j} p_{\pi(j)}\right).$$

  Furthermore, the equality can not occur in both inequalities at the same time, otherwise

$$(l - 1 = l') \wedge (r = r' + 1) \Rightarrow r - l = r' - l'\,.$$

  Thus, $\delta_l^d > \delta_{r'}^{d+1} \ge 0$, that contradicts (65).

□

Proof of Proposition 2. Lemma 1 implies that $\mathbf{K}_2 := \langle\!\langle 1, K \rangle\!\rangle$ is an optimal 2-selection. At each iterative $d = 3, \dots, K$, Alg. 1 iteratively looks for the optimal d-selection which is either the extensions $\langle\!\langle l+1, r \rangle\!\rangle$ or $\langle\!\langle l, r-1 \rangle\!\rangle$ of the optimal $(d-1)$-selection $\mathbf{K}_{d-1} := \langle\!\langle l, r \rangle\!\rangle$ as claimed in the lemma 2. The BOP is simply the optimal d-selection minimizing the generalized rank loss in expectation.

□

d	$D(\hat{\boldsymbol{y}}^d)$	$\mathbf{E}(\pi_{D_d})$	$g(K-d)$	$\mathbf{E}_d$
0	$\emptyset$	0	$c \cdot 4$	$c \cdot 4$
2	$\{1,4\}$	0.03	$c \cdot 2$	$0.03 + c \cdot 2$
3	$\{1,2,4\}$	0.17	c	$0.17 + c$
4	$\{1,2,3,4\}$	0.47	0	0.47

Table 4: BOP rank information

Proof of Remark 2. The proof is carried out with a counter example. Let $K = 4$ and $\boldsymbol{x}$ be a query instance with the conditional probabilities and the corresponding degrees of uncertainty

$$\boldsymbol{p_x} = (0.9, 0.8, 0.7, 0.3)\,, u_{\boldsymbol{x}} = (0.2, 0.4, 0.6, 0.6)\,.$$

The generalized rank loss (29) is specified by $g(|A(\hat{\boldsymbol{y}})|) := |A(\hat{\boldsymbol{y}})| \cdot c$. The information given by running the algorithm 1 is presented in Table 4. Let the cost of abstention $c := 0.03$, thus the BOP rank is $\{1, 4\}$. The BOP is clearly not uncertainty-aligned since we include the 4^{th} label with the degree of uncertainty of 0.6 while abstain the second label with degree of uncertainty of 0.4.

□

Appendix C. Proofs for Section 6 (Subset 0/1 Loss)

Proof of Proposition 3. Let π be the permutation that sorts the labels in increasing order of the degree of uncertainty (24). It is easy to check that, $\forall\, d = 0, 1, \dots, K$, $\hat{\boldsymbol{y}}^d$ is specified by the index set

$$D(\hat{\boldsymbol{y}}^d) := \{k \in [K] \,|\, \pi^{-1}(k) \leq d\}\,,$$

and $\forall\, k \in D(\hat{\boldsymbol{y}}^d)$

$$\hat{y}_k^d = \operatorname*{argmin}_{\bar{y}_k \in \{0,1\}} p_k^{1-\bar{y}_k}(1-p_k)^{\bar{y}_k}\,.$$

Note that, $\forall\, \bar{\boldsymbol{y}} \in \mathcal{Y}_d^*$, we have constant penalty $g(K-d) := g(|A(\bar{\boldsymbol{y}})|)$. Thus

$$\begin{aligned}
\mathbf{E}\left(\ell_S(\boldsymbol{y}_D, \bar{\boldsymbol{y}}_D)\right) &= 1 - \boldsymbol{p}(\bar{\boldsymbol{y}} \,|\, \boldsymbol{x}) = 1 - \prod_{k \in D(\bar{\boldsymbol{y}})} p_k^{\bar{y}_k}(1-p_k)^{1-\bar{y}_k}\,,\\
&\geq 1 - \prod_{k \in D(\bar{\boldsymbol{y}})} \max_{\bar{y}_k \in \{0,1\}} p_k^{\bar{y}_k}(1-p_k)^{1-\bar{y}_k}\\
&\geq 1 - \prod_{k \in D(\hat{\boldsymbol{y}}^d)} \max_{\bar{y}_k \in \{0,1\}} p_k^{\bar{y}_k}(1-p_k)^{1-\bar{y}_k}\\
&= \mathbf{E}\left(\ell(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D^d)\right)\,.
\end{aligned}$$

The second inequality holds because replacing any $k \in D(\hat{\boldsymbol{y}}^d)$ by $k' \in [K] \setminus D(\hat{\boldsymbol{y}}^d)$ increase

$$\min_{\bar{y}_k \in \{0,1\}} p_k^{1-\bar{y}_k}(1-p_k)^{\bar{y}_k}\,,$$

or equivlently decreases

$$\max_{\bar{y}_k \in \{0,1\}} p_k^{\bar{y}_k}(1-p_k)^{1-\bar{y}_k} .$$

The proof is complete by the fact that $\forall\, k \in [K]$, the optimal prediction is

$$\begin{aligned}\hat{y}_k^d &= \operatorname*{argmax}_{\bar{y}_k \in \{0,1\}} p_k^{\bar{y}_k}(1-p_k)^{1-\bar{y}_k} \\ &= \operatorname*{argmin}_{\bar{y}_k \in \{0,1\}} p_k^{1-\bar{y}_k}(1-p_k)^{\bar{y}_k} .\end{aligned}$$

□

Appendix D. Proofs for Section 7 (Confusion Matrix-derived Accuracy Measures)

Proof of Proposition 4. The *monotonicity* introduced in Definition 1 can be rewritten for the generalization F (29) of an accuracy measure f as follows: F should only increase (or at least not decrease) in the following cases

(D1) turning an incorrect prediction on a label λ_k into a correct prediction,

(D2) turning an abstention on a label λ_k into a correct prediction,

(D3) turning an incorrect prediction on a label λ_k into an abstention.

We first show that (D1) and (D2) are always ensured given g is a non-decreasing function of $|A(\hat{\boldsymbol{y}})|$. Once this is done, the proof is reduced to determining the conditions under which (D3) is (not) satisfied.

- (D1) Turning an incorrect prediction into a correct prediction either means correcting a false positive or a false negative. Let us remind that, for any pair $(\boldsymbol{y}, \hat{\boldsymbol{y}}) \in \mathcal{Y} \times \mathcal{Y}^*$, we have that

  $$F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) - g(|A(\hat{\boldsymbol{y}})|).$$

  Thus, in the first case, $F(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is replaced by

  $$f_{tp,fp}^{+1,-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) - g(|A(\hat{\boldsymbol{y}})|) ,$$

  in the second case by

  $$f_{tn,fn}^{+1,-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) - g(|A(\hat{\boldsymbol{y}})|) .$$

  In both cases, the value of the measure F increases i.e.,

  $$f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) - g(|A(\hat{\boldsymbol{y}})|) \leq \min\left(f_{tp,fp}^{+1,-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D), f_{tn,fn}^{+1,-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right) - g(|A(\hat{\boldsymbol{y}})|) ,$$

  since (D4) and (D5) of Definition 4 ensure that

  $$f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) \leq \min\left(f_{tp,fp}^{+1,-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D), f_{tn,fn}^{+1,-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right) .$$

- (D2) Turning an abstention into a correct prediction either means adding a true positive or a true negative while reducing the size of the abstention set by 1. In the first case, $F(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is replaced by

  $$f_{tp}^{+1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) - g(|A(\hat{\boldsymbol{y}})| - 1)\,,$$

  in the second case by

  $$f_{tn}^{+1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) - g(|A(\hat{\boldsymbol{y}})| - 1)\,.$$

  In both cases, the value of the measure F increases i.e.,

  $$f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) - g(|A(\hat{\boldsymbol{y}})|) \leq \min\left(f_{tp}^{+1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D), f_{tn}^{+1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right) - g(|A(\hat{\boldsymbol{y}}| - 1)\,,$$

  since g is a non-decreasing function of $|A(\hat{\boldsymbol{y}}|$ and f is monotone increasing in *tp* and *tn*.

(⇒) We are going to show that if F (29) is monotonic in the sense of Definition 1, then the following condition is satisfied:

$$g(|A(\hat{\boldsymbol{y}})| + 1) - g(|A(\hat{\boldsymbol{y}})|) \leq \min\left(f_{fp}^{-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D), f_{fn}^{-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right) - f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\,,$$

for any pair $(\boldsymbol{y}, \hat{\boldsymbol{y}}) \in \mathcal{Y} \times \mathcal{Y}^*$. Since (D1) and (D2) are always ensured given g is a non-decreasing function of $|A(\hat{\boldsymbol{y}})|$, it remains to determine the conditions under which (D3) is satisfied. Turning an incorrect prediction into an abstention either means subtracting a false positive or a false negative while increasing the size of the abstention set by 1. In the first case, $F(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is replaced by

$$f_{fp}^{-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) - g(|A(\hat{\boldsymbol{y}})| + 1)\,,$$

in the second case by

$$f_{fn}^{-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) - g(|A(\hat{\boldsymbol{y}})| + 1)\,.$$

In both cases, the monotonicity of F as given by (29) implies that the value of the measure F should increase. Thus,

$$f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D) - g(|A(\hat{\boldsymbol{y}})|) \leq \min\left(f_{fp}^{-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D), f_{fn}^{-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right) - g(|A(\hat{\boldsymbol{y}})| + 1)\,,$$

or equivalently,

$$g(|A(\hat{\boldsymbol{y}})| + 1) - g(|A(\hat{\boldsymbol{y}})|) \leq \min\left(f_{fp}^{-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D), f_{fn}^{-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right) - f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\,.$$

(⇐) To this end, we show that if

$$g(|A(\hat{\boldsymbol{y}})| + 1) - g(|A(\hat{\boldsymbol{y}})|) \leq \min\left(f_{fp}^{-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D), f_{fn}^{-1}(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right) - f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)$$

for any pair $(\boldsymbol{y}, \hat{\boldsymbol{y}}) \in \mathcal{Y} \times \mathcal{Y}^*$, then (D3) is satisfied, which in turn implies that F (29) is monotonic in the sense of Definition 1. This is obvious, because the above condition on f and g ensures that turning an incorrect prediction into an abstention (i.e., either subtracting a false positive or a false negative while increasing the size of the abstention set by 1) increases the value of the measure F as defined in (29). □

Proof of Remark 3. Assume the marginal probabilities p_k to be given, and (conditional) independence of label probabilities in the sense of (15) to hold. Let f be an MLC accuracy measure which is a monotone function of *tp*, *tn*, *fp* and *fn* as described in Remark 3. We start by showing that, for any partial prediction $\hat{\boldsymbol{y}} \in \mathcal{Y}$ and any index $k \in [K]$,

(R1) $\mathbf{E}\left(f(\boldsymbol{y}, \hat{\boldsymbol{y}})\right)$ is monotone increasing in p_k if $\hat{y}_k = 1$;

(R2) $\mathbf{E}\left(f(\boldsymbol{y}, \hat{\boldsymbol{y}})\right)$ is monotone decreasing in p_k if $\hat{y}_k = 0$.

To this end, fix some $k \in [K]$ and denote by $\boldsymbol{y}_{-k} := (y_1, \dots, y_{k-1}, y_{k+1}, \dots, y_K)$ the labeling induced from $\boldsymbol{y} \in \mathcal{Y}$ by removing the k^{th} element. Moreover, we introduce the shorthand notation

$$\alpha(\boldsymbol{y}) := \prod_{k \in [K]} p_k^{y_k}(1-p_k)^{1-y_k} .$$

- (R1) Let $\mathcal{Y}_{-k} := \{\boldsymbol{y}_{-k} \,|\, \boldsymbol{y} \in \mathcal{Y}\}$. We consider the case where $\hat{y}_k = 1$. We can separate the cases $\boldsymbol{y} \in \mathcal{Y}$ with $y_k = 1$ from those with $y_k = 0$, which yields

$$\begin{aligned}\mathbf{E}\left(f(\boldsymbol{y}, \hat{\boldsymbol{y}})\right) &= \sum_{\boldsymbol{y} \in \mathcal{Y} : y_k = 1} f(\boldsymbol{y}, \hat{\boldsymbol{y}}) \boldsymbol{p}(\boldsymbol{y} \,|\, \boldsymbol{x}) + \sum_{\boldsymbol{y} \in \mathcal{Y} : y_k = 0} f(\boldsymbol{y}, \hat{\boldsymbol{y}}) \boldsymbol{p}(\boldsymbol{y} \,|\, \boldsymbol{x}) \\ &= \sum_{\boldsymbol{y}_{-k} \in \mathcal{Y}_{-k}} p_k \alpha(\boldsymbol{y}_{-k}) f_{tp}^{+1}(\boldsymbol{y}_{-k}, \hat{\boldsymbol{y}}_{-k}) + \sum_{\boldsymbol{y}_{-k} \in \mathcal{Y}_{-k}} (1-p_k) \alpha(\boldsymbol{y}_{-k}) f_{fn}^{+1}(\boldsymbol{y}_{-k}, \hat{\boldsymbol{y}}_{-k}) .\end{aligned}$$

  Because f is monotone increasing in *tp* and monotone decreasing in *fn*, we have

$$f_{tp}^{+1}(\boldsymbol{y}_{-k}, \hat{\boldsymbol{y}}_{-k}) \geq f_{fn}^{+1}(\boldsymbol{y}_{-k}, \hat{\boldsymbol{y}}_{-k}) .$$

  Thus, $\mathbf{E}\left(f(\boldsymbol{y}, \hat{\boldsymbol{y}})\right)$ is monotone increasing in p_k.

- (R2) We consider the case where $\hat{y}_k = 0$. Separating the cases $\boldsymbol{y} \in \mathcal{Y}$ with $y_k = 1$ from those with $y_k = 0$, we have

$$\begin{aligned}\mathbf{E}\left(f(\boldsymbol{y}, \hat{\boldsymbol{y}})\right) &= \sum_{\boldsymbol{y} \in \mathcal{Y} : y_k = 1} f(\boldsymbol{y}, \hat{\boldsymbol{y}}) \boldsymbol{p}(\boldsymbol{y} \,|\, \boldsymbol{x}) + \sum_{\boldsymbol{y} \in \mathcal{Y} : y_k = 0} f(\boldsymbol{y}, \hat{\boldsymbol{y}}) \boldsymbol{p}(\boldsymbol{y} \,|\, \boldsymbol{x}) \\ &= \sum_{\boldsymbol{y}_{-k} \in \mathcal{Y}_{-k}} p_k \alpha(\boldsymbol{y}_{-k}) f_{fp}^{+1}(\boldsymbol{y}_{-k}, \hat{\boldsymbol{y}}_{-k}) + \sum_{\boldsymbol{y}_{-k} \in \mathcal{Y}_{-k}} (1-p_k) \alpha(\boldsymbol{y}_{-k}) f_{tn}^{+1}(\boldsymbol{y}_{-k}, \hat{\boldsymbol{y}}_{-k}) .\end{aligned}$$

  Because f is monotone increasing in *tn* and monotone decreasing in *fp*, we have

$$f_{fp}^{+1}(\boldsymbol{y}_{-k}, \hat{\boldsymbol{y}}_{-k}) \leq f_{tn}^{+1}(\boldsymbol{y}_{-k}, \hat{\boldsymbol{y}}_{-k}) .$$

  Thus, $\mathbf{E}\left(f(\boldsymbol{y}, \hat{\boldsymbol{y}})\right)$ is monotone decreasing in p_k.

We prove the second part of the remark by contradiction. So, contrary to the claim, suppose that f has no BOP of the form (56). Then, there are indices i, j s.t. $\hat{y}_i = 1$, $\hat{y}_j = 0$, and $p_j > p_i$. However, since $\mathbf{E}\left(f(\boldsymbol{y}, \hat{\boldsymbol{y}})\right)$ is an increasing function of p_i and a decreasing function of p_j according to (R1) and (R2), it is at least not decreasing when replacing i by j. This contradicts the assumption. □

Proof of Lemma 3. Note that for any $\hat{y} \in \mathcal{Y}_d^*$, $A(\hat{y}) = K - d$ is a constant. Therefore, $\mathbf{E}\left(F(\boldsymbol{y}, \hat{\boldsymbol{y}})\right)$ is increasing (decreasing) iff $\mathbf{E}\left(f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right)$ is increasing (decreasing).

Using Remark 3, we can easily verify that for any partial prediction $\hat{\boldsymbol{y}} \in \mathcal{Y}^*$ and any index $k \in D(\hat{\boldsymbol{y}})$, (R3) $\mathbf{E}\left(f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right)$ is monotone increasing in p_k if $\hat{y}_k = 1$;

(R4) $\mathbf{E}\left(f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right)$ is monotone decreasing in p_k if $\hat{y}_k = 0$.

Thus, for any $i, j \in D(\hat{\boldsymbol{y}}^d)$ s.t. $\hat{y}_i = 1$, $\hat{y}_j = 0$, we should have $p_i > p_j$. Otherwise, swapping the predictions on y_i and y_j increases $\mathbf{E}\left(f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right)$, which leads to a contradiction. Now, suppose that, contrary to the claim of the lemma, the solution $\hat{\boldsymbol{y}}^d$ of the inner maximization (52) cannot be expressed by a decision set of the form $\langle\!\langle l, r \rangle\!\rangle$. Then, for the optimal solution, we have at least one of the following cases:

(i) $\exists i \in D(\hat{\boldsymbol{y}}^d)$, $j \in A(\hat{\boldsymbol{y}}^d)$ s.t. $\hat{y}_i = 0$, $\hat{y}_j = \bot$, and $p_j < p_i$,

(ii) $\exists i \in D(\hat{\boldsymbol{y}}^d)$, $j \in A(\hat{\boldsymbol{y}}^d)$ s.t. $\hat{y}_i = 1$, $\hat{y}_j = \bot$, and $p_j < p_i$.

The proof is completed by showing that both (i) and (ii) lead to a contradiction:

(i) Suppose $\hat{y}_i = 1$, $\hat{y}_j = \bot$, and $p_j > p_i$. According to (R3), $\mathbf{E}\left(f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right)$ is an increasing function of p_i. Therefore, this value does at least not decrease when swapping the predictions on y_i and y_j, which is a contradiction.

(ii) Suppose $\hat{y}_i = 0$, $\hat{y}_j = \bot$, and $p_j < p_i$. According to (R4), $\mathbf{E}\left(f(\boldsymbol{y}_D, \hat{\boldsymbol{y}}_D)\right)$ is a decreasing function of p_i. Therefore, this value does at least not decrease when swapping the predictions on y_i and y_j, which is a contradiction.

□

Proof of Proposition 5. The proof is obvious and immediately follows from the previous lemma, because the BOP is given by the optimal partial prediction in $\mathcal{Y}^*_{\hat{d}}$, where

$$\hat{d} := \operatorname*{argmax}_{d=0,1,\ldots,K} \mathbf{E}\left(F(\boldsymbol{y}, \hat{\boldsymbol{y}}^d)\right) .$$

□

Proof of Lemma 4. To compute $Q(l, l_1)$, $0 \le l_1 \le l \le K$, we employ a $K \times (K+2)$ matrix $\boldsymbol{Q}$, with $l_1 = -1, 0 \ldots, K+1$ and $l = 1, \ldots, K$, and update it via dynamic programming (similar to the procedure discussed by Decubber et al. (2018) using double indexing, with $l_1 = -1, 0 \ldots, l+1$).

At the beginning, all elements of $\boldsymbol{Q}$ are initialized by 0, except the first row:

$$Q(1, \cdot) = \left(0, 1 - p_{(1)}, p_{(1)}, 0, \ldots, 0\right) .$$

We then iteratively update the rows $l = 2, 3, \ldots, K$ of $\boldsymbol{Q}$ in a dynamic programming style:

$$Q(l, l_1) = p_{(l)} Q(l-1, l_1 - 1) + \left(1 - p_{(l)}\right) Q(l-1, l_1) ,$$

with $l_1 = 0, 1, \ldots, l$.

Let $r' := K + 1 - r$ and

$$P(r', r'_1) := \boldsymbol{p}\left(\sum_{k=r}^{K} y_{\pi(k)} = r'_1 \,\middle|\, \boldsymbol{x}\right) .$$

We can compute $P(r', r'_1)$, $0 \leq r'_1 \leq r' \leq K$, in a way similar to the computation of $Q(l, l_1)$. We employ a $K \times (K+2)$ matrix P, with $r'_1 = -1, 0, \ldots, K+1$, and $r' = 1, 2, \ldots, K$. At the beginning, all elements are initialized by 0, except the first row:

$$P(1, \cdot) = \left(0, 1 - p_{(K)}, p_{(K)}, 0, \ldots, 0\right) .$$

We then iteratively update the rows $r = 2, 3, \ldots, K$ via dynamic programming:

$$P(r', r'_1) = p_{(r)} P(r'-1, r'_1 - 1) + \left(1 - p_{(r)}\right) P(r'-1, r'_1) ,$$

with $r'_1 = 0, 1, \ldots, r'$.

□

Proof of Proposition 6. Let π be the permutation sorts the labels in decreasing order of the marginal probabilities p_k, and assume CLI in the sense of (15). In the following, we show that a BOP of the generalized measure F_β (29) can be constructed in time $O(K^3)$. Denoting by $\beta' := 1 + \beta^{-2}$ and using the shorthand notation

$$S_\beta(l, l_1, r') := \sum_{r'_1=0}^{r'} \frac{P(r', r'_1)}{l\beta^{-2} + l_1 + r'_1} ,$$

the expectation of the generalized measure F_β of $\hat{\boldsymbol{y}}_r^l$ is

$$\begin{aligned}
F_\beta(l, r) &= \sum_{l_1=0}^{l} \sum_{r'_1=0}^{r'} \frac{\beta' l_1 Q(l, l_1) P(r', r'_1)}{l + \beta^2(l_1 + r'_1)} - g(r - l - 1) \\
&= \beta' \sum_{l_1=0}^{l} l_1 Q(l, l_1) \sum_{r'_1=0}^{r'} \frac{P(r', r'_1)}{l\beta^{-2} + l_1 + r'_1} - g(r - l - 1) \\
&= \beta' \sum_{l_1=0}^{l} l_1 Q(l, l_1) S(l, l_1, r') - g(r - l - 1) .
\end{aligned}$$

$S_\beta(l, l_1, r')$ can be computed recursively as follows:

$$\begin{aligned}
S_\beta(l, l_1, r') &= p_{\pi(r)} \sum_{r'_1=1}^{r'} \frac{\boldsymbol{p}\left(\sum_{k=r+1}^{K} y_{\pi(k)} = r'_1 - 1 | \boldsymbol{x}\right)}{l\beta^{-2} + l_1 + r'_1} + \left(1 - p_{\pi(r)}\right) \sum_{r'_1=0}^{r'-1} \frac{\boldsymbol{p}\left(\sum_{k=r+1}^{K} y_{\pi(k)} = r'_1 | \boldsymbol{x}\right)}{l\beta^{-2} + l_1 + r'_1} \\
&= p_{\pi(r)} \sum_{r'_1=0}^{r'-1} \frac{\boldsymbol{p}\left(\sum_{k=r+1}^{K} y_{\pi(k)} = r'_1 | \boldsymbol{x}\right)}{l\beta^{-2} + (l_1 + 1) + r'_1} + \left(1 - p_{\pi(r)}\right) \sum_{r'_1=0}^{r'-1} \frac{\boldsymbol{p}\left(\sum_{k=r+1}^{K} y_{\pi(k)} = r'_1 | \boldsymbol{x}\right)}{l\beta^{-2} + l_1 + r'_1} \\
&= p_{\pi(r)} S(l, l_1 + 1, r' - 1) + \left(1 - p_{\pi(r)}\right) S_\beta(l, l_1, r' - 1) ,
\end{aligned}$$

with the boundary conditions

$$S_\beta(l, l_1, 0) = \frac{1}{k\beta^{-2} + l_1} , \forall (l, l_1) .$$

Altogether, we end up to the implementation given in Algorithm 2, which has time complexity $O(K)^3$.

□

Proof of Proposition 7. Using the shorthand notation

$$S_{\mathrm{Jac}}(l, r') := \sum_{r'_1=0}^{r'} \frac{P(r', r'_1)}{l + r'_1} \,,$$

the expectation of the generalized measure F_{Jac} of $\hat{\boldsymbol{y}}_r^l$ is

$$\begin{aligned}
F_{\mathrm{Jac}}(l, r) &= \sum_{l_1=0}^{l} \sum_{r'_1=0}^{r'} \frac{l_1 Q(l, l_1) P(r', r'_1)}{l + r'_1} - g(r - l - 1) \\
&= \sum_{l_1=0}^{l} l_1 Q(l, l_1) \sum_{r'_1=0}^{r'} \frac{P(r', r'_1)}{l + r'_1} - g(r - l - 1) \\
&= \sum_{l_1=0}^{l} l_1 Q(l, l_1) S_{\mathrm{Jac}}(l, r') - g(r - l - 1) \\
&= S_{\mathrm{Jac}}(l, r') \sum_{l_1=0}^{l} l_1 Q(l, l_1) - g(r - l - 1) \,.
\end{aligned}$$

$S_{\mathrm{Jac}}(l, r')$ can be computed recursively as follows:

$$\begin{aligned}
S_{\mathrm{Jac}}(l, r') &= p_{(r)} \sum_{r'_1=1}^{r'} \frac{\boldsymbol{p}\left(\sum_{k=r+1}^{K} y_{\pi(k)} = r'_1 - 1 | \boldsymbol{x}\right)}{l + r'_1} + \left(1 - p_{\pi(r)}\right) \sum_{r'_1=0}^{r'-1} \frac{\boldsymbol{p}\left(\sum_{k=r+1}^{K} y_{\pi(k)} = r'_1 | \boldsymbol{x}\right)}{l + r'_1} \\
&= p_{\pi(r)} \sum_{r'_1=0}^{r'-1} \frac{\boldsymbol{p}\left(\sum_{k=r+1}^{K} y_{\pi(k)} = r'_1 | \boldsymbol{x}\right)}{(l + 1) + r'_1} + \left(1 - p_{\pi(r)}\right) \sum_{r'_1=0}^{r'-1} \frac{\boldsymbol{p}\left(\sum_{k=r+1}^{K} y_{\pi(k)} = r'_1 | \boldsymbol{x}\right)}{l + r'_1} \\
&= p_{\pi(r)} S_{\mathrm{Jac}}(l + 1, r' - 1) + \left(1 - p_{\pi(r)}\right) S_{\mathrm{Jac}}(l, r' - 1) \,,
\end{aligned}$$

with the boundary conditions

$$S(l, 0) = \frac{1}{l} \,, \forall l \,.$$

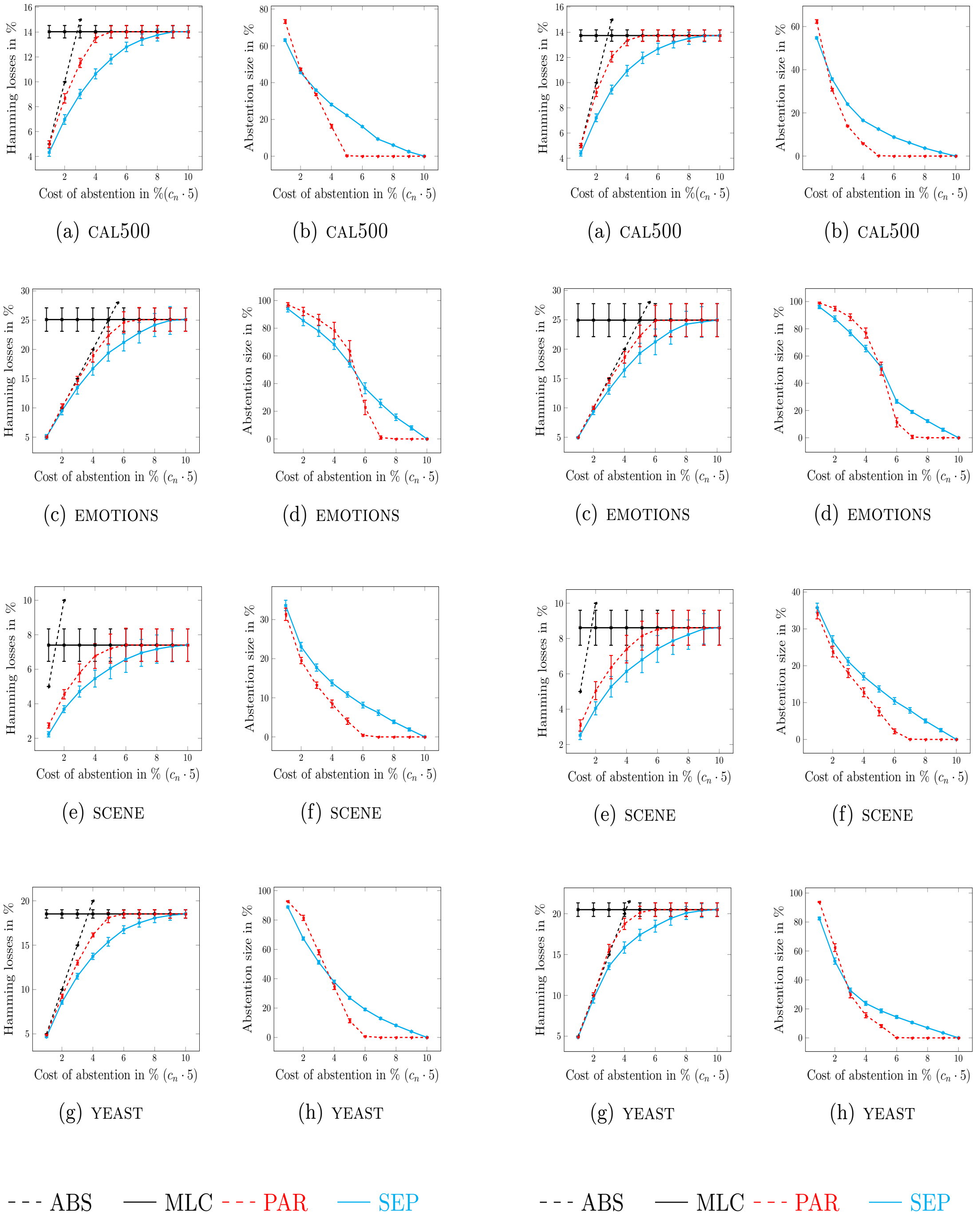
Altogether, we end up to the implementation given in Algorithm 3, which has time complexity $O(K)^3$. □

Appendix E. Additional Experiments

In addition to the experiments which presented in Section 9, we also conduct experiments with BR+SVM and ECC+SVM. Note that the standard SVMs do not provide probabilistic predictions, we train the parameters of an additional sigmoid function to map the SVM outputs into probabilities (Lin et al., 2007; Platt, 1999) using an internal five-fold cross validation. Similar results (to the ones presented in Section 9) are given for the Hamming loss (c.f. fig. 8), rank loss (c.f. fig. 9), Subset 0/1 Loss (c.f. fig. 10), F_1-measure (c.f. fig. 11) and Jaccard Measure (c.f. fig. 12).

Appendix F. Deep Neural Network Training

Details of the parameter setting for the VGG16-based convolutional neural network classifier are given in Figure 13.



(a) CAL500 (b) CAL500 (a) CAL500 (b) CAL500

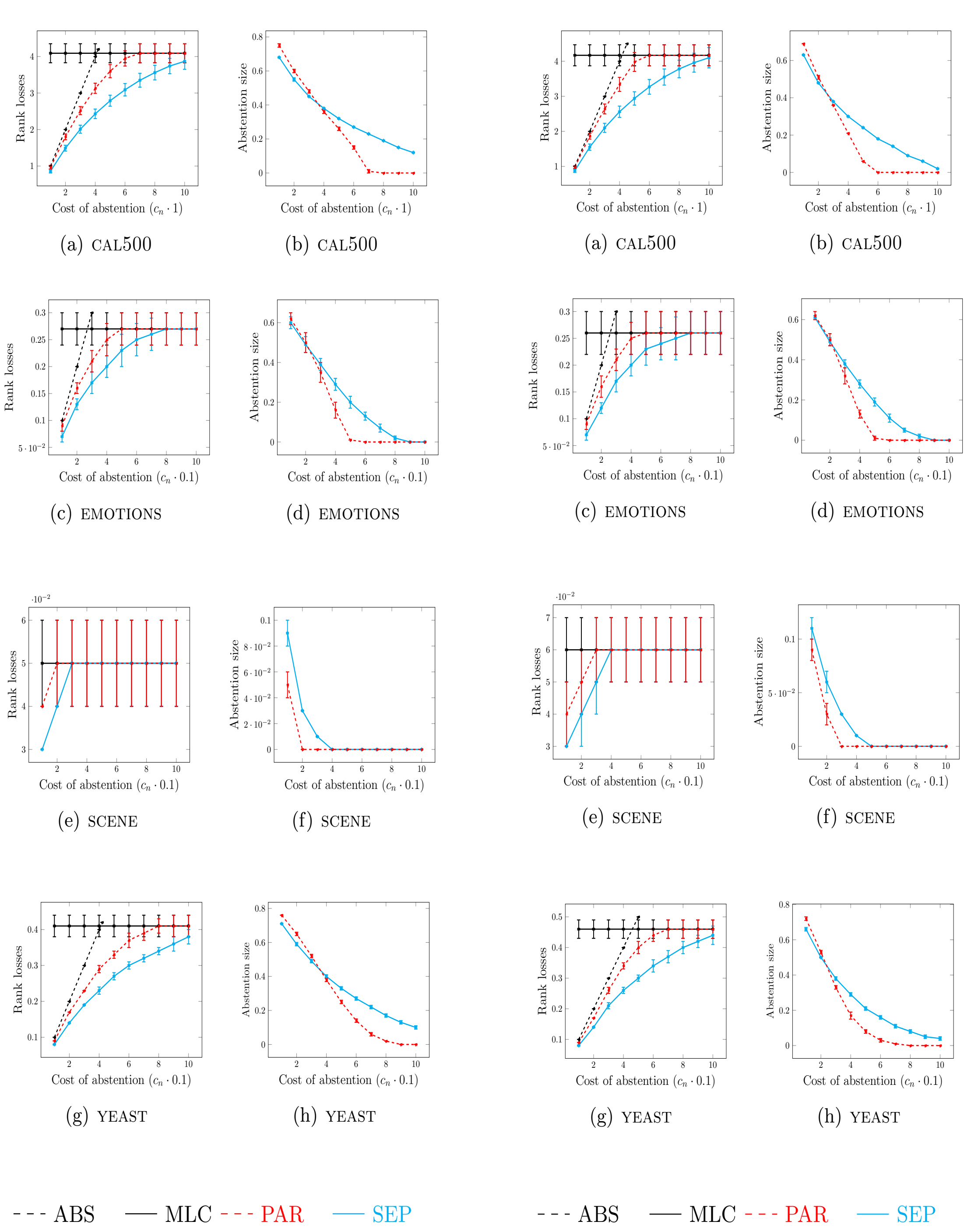
(c) EMOTIONS (d) EMOTIONS (c) EMOTIONS (d) EMOTIONS

(e) SCENE (f) SCENE (e) SCENE (f) SCENE

(g) YEAST (h) YEAST (g) YEAST (h) YEAST

a. BR+SVM b. ECC+SVM

Figure 8: Experimental results in terms of average Hamming loss (in percent), which is plotted in percent of the maximal loss K, and abstention size (in percent) for $g_1(a) = a \cdot c$ (SEP) and $g_2(a) = (a \cdot K \cdot c)/(K + a)$ (PAR), as a function of the cost of abstention.



(a) CAL500 (b) CAL500 (a) CAL500 (b) CAL500

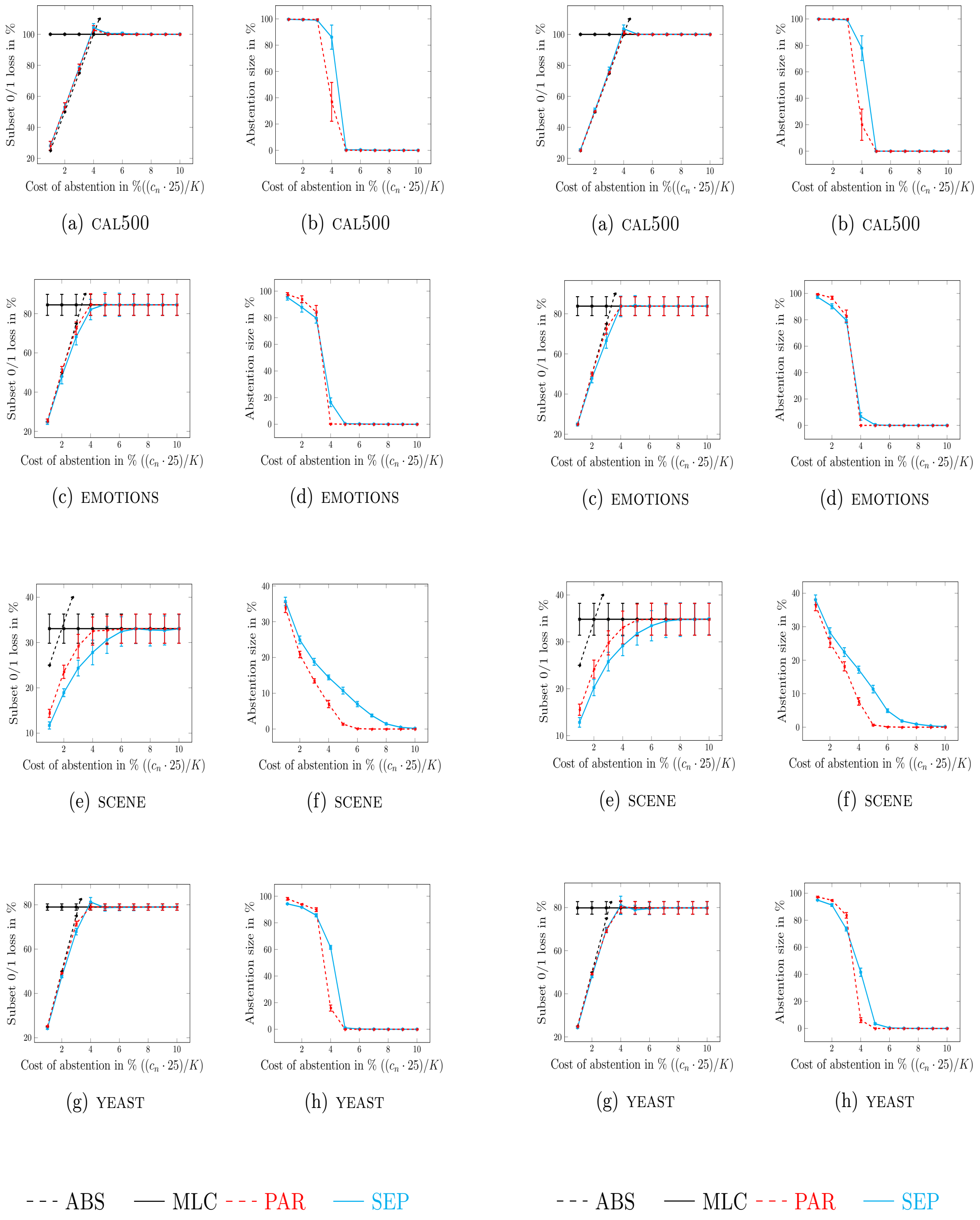
(c) EMOTIONS (d) EMOTIONS (c) EMOTIONS (d) EMOTIONS

(e) SCENE (f) SCENE (e) SCENE (f) SCENE

(g) YEAST (h) YEAST (g) YEAST (h) YEAST

a. BR+SVM b. ECC+SVM

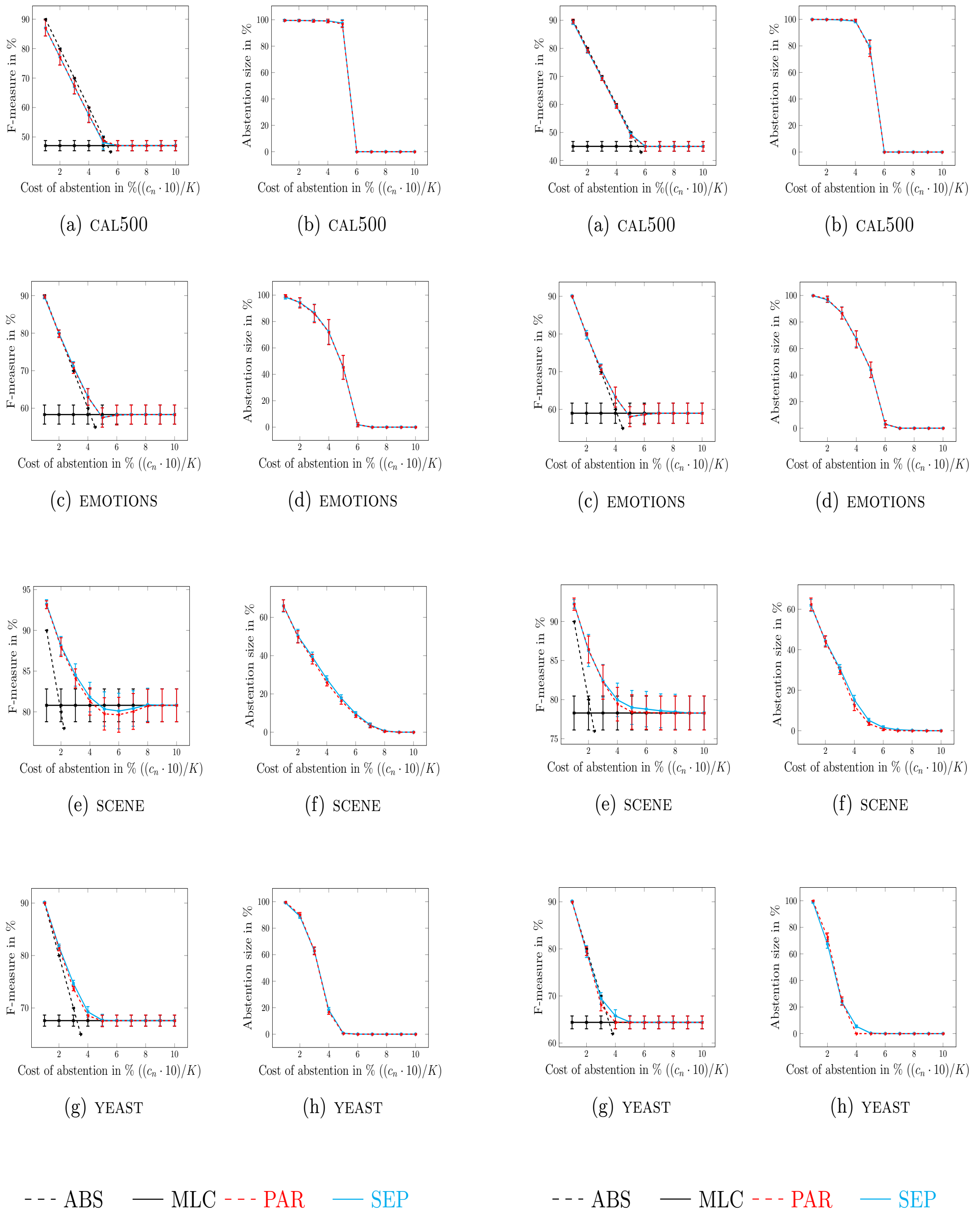
Figure 9: Experimental results in terms of average rank loss L_R and abstention size for $g_1(a) = a \cdot c$ (SEP) and $g_2(a) = (a \cdot K \cdot c)/(K + a)$ (PAR), as a function of the cost of abstention.



a. BR+SVM

b. ECC+SVM

Figure 10: Experimental results in terms of average Subset 0/1 loss L_S (in percent) and abstention size (in percent) for $g_1(a) = a \cdot c$ (SEP) and $g_2(a) = (a \cdot K \cdot c)/(K + a)$ (PAR), as a function of the cost of abstention.



(a) CAL500 (b) CAL500 (a) CAL500 (b) CAL500

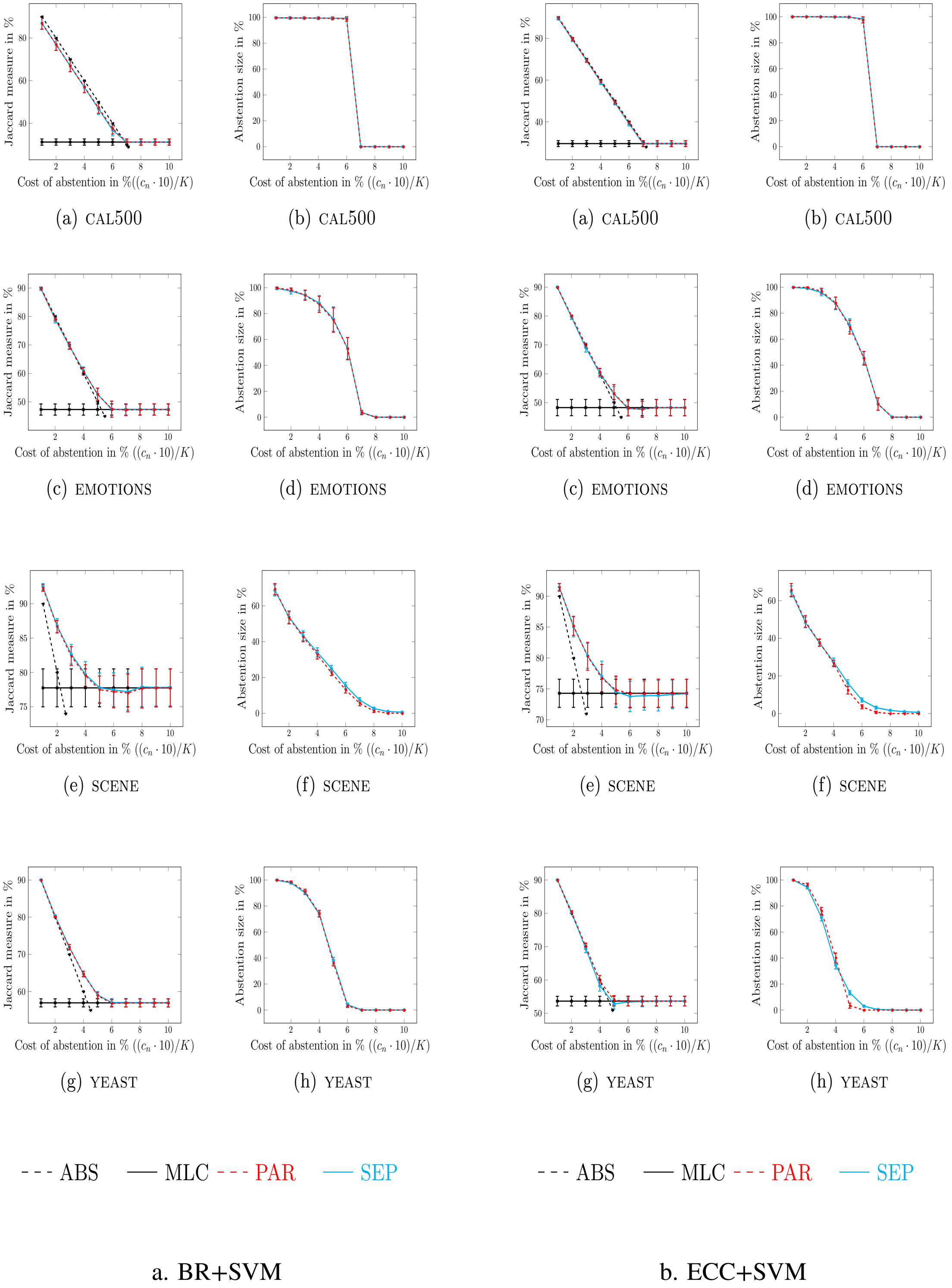
(c) EMOTIONS (d) EMOTIONS (c) EMOTIONS (d) EMOTIONS

(e) SCENE (f) SCENE (e) SCENE (f) SCENE

(g) YEAST (h) YEAST (g) YEAST (h) YEAST

a. BR+SVM b. ECC+SVM

Figure 11: Experimental results in terms of average F_1 (in percent) and abstention size (in percent) for $g_1(a) = a \cdot c$ (SEP) and $g_2(a) = (a \cdot K \cdot c)/(K + a)$ (PAR), as a function of the cost of abstention.



(a) CAL500 (b) CAL500 (a) CAL500 (b) CAL500

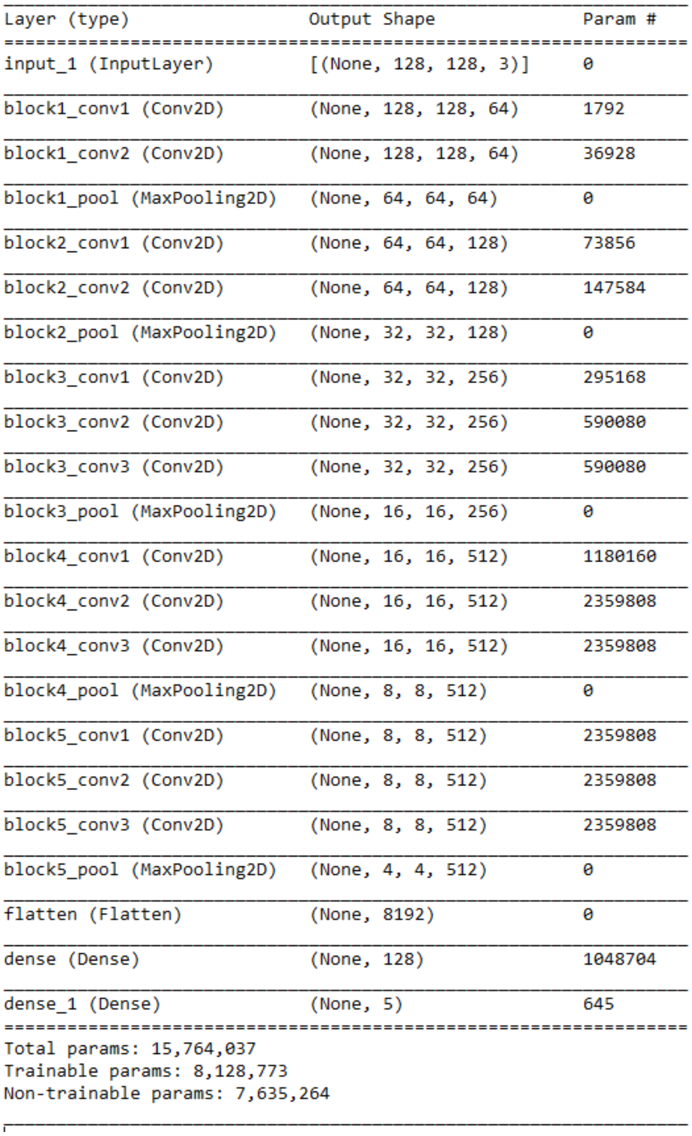
(c) EMOTIONS (d) EMOTIONS (c) EMOTIONS (d) EMOTIONS

(e) SCENE (f) SCENE (e) SCENE (f) SCENE

(g) YEAST (h) YEAST (g) YEAST (h) YEAST

a. BR+SVM b. ECC+SVM

Figure 12: Experimental results in terms of average Jaccard measure F_{Jac} (in percent) and abstention size (in percent) for $g_1(a) = a \cdot c$ (SEP) and $g_2(a) = (a \cdot K \cdot c)/(K + a)$ (PAR), as a function of the cost of abstention.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 128, 128, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 128, 128, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 128, 128, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 64, 64, 64)        0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 64, 64, 128)       73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 64, 64, 128)       147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 32, 32, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 32, 32, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 32, 32, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 32, 32, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 16, 16, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 16, 16, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 8, 8, 512)         0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 8, 8, 512)         2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 8, 8, 512)         2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 8, 8, 512)         2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 4, 4, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 8192)              0
_________________________________________________________________
dense (Dense)                (None, 128)               1048704
_________________________________________________________________
dense_1 (Dense)              (None, 5)                 645
=================================================================
Total params: 15,764,037
Trainable params: 8,128,773
Non-trainable params: 7,635,264
_________________________________________________________________
```

Figure 13: Parameter setting for the VGG16-based convolutional neural network.

References

Antonucci, A. and Corani, G. (2017). The multilabel naive credal classifier. *International Journal of Approximate Reasoning*, 83:320–336.

Bartlett, P. L. and Wegkamp, M. H. (2008). Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(Aug):1823–1840.

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

Chai, K. M. A. (2005). Expectation of f-measures: Tractable exact computation and some empirical observations of its properties. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, pages 593–594. ACM.

Cheng, W., Hüllermeier, E., Waegeman, W., and Welker, V. (2012). Label ranking with partial abstention based on thresholded probabilistic models. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2501–2509.

Cheng, W., Rademaker, M., De Baets, B., and Hüllermeier, E. (2010). Predicting partial orders: ranking with abstention. In *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part I (ECML/PKDD)*, pages 215–230. Springer-Verlag.

Chow, C. (1970). On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1):41–46.

Cortes, C., DeSalvo, G., and Mohri, M. (2016). Learning with rejection. In *Proceedings of the 27th International Conference on Algorithmic Learning Theory (ALT)*, pages 67–82. Springer Verlag.

Decubber, S., Mortier, T., Dembczyński, K., and Waegeman, W. (2018). Deep f-measure maximization in multi-label classification: A comparative study. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, pages 290–305. Springer.

Dembczyński, K., Waegeman, W., Cheng, W., and Hüllermeier, E. (2012). On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1-2):5–45.

Destercke, S. (2015). Multilabel predictions with sets of probabilities: The hamming and ranking loss cases. *Pattern Recognition*, 48(11):3757–3765.

Elisseeff, A. and Weston, J. (2001). A kernel method for multi-labelled classification. In *Proceedings of the 14th International Conference on Neural Information Processing Systems (NIPS)*, pages 681–687. MIT Press.

Fan, R.-E. and Lin, C.-J. (2007). A study on threshold selection for multi-label classification.

Franc, V. and Prusa, D. (2019). On discriminative learning of prediction uncertainty. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 1963–1971.

Grandvalet, Y., Rakotomamonjy, A., Keshet, J., and Canu, S. (2008). Support vector machines with a reject option. In *Proceedings of the 21st International Conference on Neural Information Processing Systems (NIPS)*, pages 537–544. Curran Associates Inc.

Hayes, P. J. and Weinstein, S. P. (1990). Construe/tis: A system for content-based indexing of a database of news stories. In *Proceedings of The Second Conference on Innovative Applications of Artificial Intelligence (IAAI)*, pages 49–64. AAAI Press.

Hellman, M. E. (1970). The nearest neighbor classification rule with a reject option. *IEEE Transactions on Systems Science and Cybernetics*, 6(3):179–185.

Jansche, M. (2007). A maximum expected utility framework for binary sequence labeling. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 736–743.

Jasinska, K., Dembczyński, K., Busa-Fekete, R., Klerx, T., and Hüllermeier, E. (2016). Extreme F-measure maximization using sparse probability estimates. In *Proceedings of the 33th International Conference on Machine Learning (ICML)*, pages 1435–1444.

Lewis, D. D. (1992). An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 37–50. ACM.

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

Lin, H.-T., Lin, C.-J., and Weng, R. C. (2007). A note on platt's probabilistic outputs for support vector machines. *Machine learning*, 68(3):267–276.

Luque, A., Carrasco, A., Martín, A., and de las Heras, A. (2019a). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231.

Luque, A., Carrasco, A., Martín, A., and Lama, J. (2019b). Exploring symmetry of binary classification performance metrics. *Symmetry*, 11(1):47.

Nguyen, V.-L., Destercke, S., Masson, M.-H., and Hüllermeier, E. (2018). Reliable multi-class classification based on pairwise epistemic and aleatoric uncertainty. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5089–5095.

Nguyen, V.-L. and Hüllermeier, E. (2020). Reliable multi-label classification: Prediction with partial abstention. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pages 5264–5271. AAAI.

Nguyen, V.-L., Hüllermeier, E., Rapp, M., Mencía, E. L., and Fürnkranz, J. (2020). On aggregation in ensembles of multilabel classifiers. In *Proceedings of the 23nd International Conference on Discovery Science (DS)*, pages 533–547. Springer.

Park, L. A. and Simoff, S. (2015). Using entropy as a measure of acceptance for multi-label classification. In *Proceedings of the 14th International Symposium on Intelligent Data Analysis (IDA)*, pages 217–228. Springer.

Pillai, I., Fumera, G., and Roli, F. (2013). Multi-label classification with a reject option. *Pattern Recognition*, 46(8):2256–2266.

Pillai, I., Fumera, G., and Roli, F. (2017). Designing multi-label classifiers that maximize f measures: State of the art. *Pattern Recognition*, 61:394–404.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classiers*, pages 1–11.

Powers, D. (2011). Evaluation: From predcision, recall and f-factor to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63.

Quevedo, J. R., Luaces, O., and Bahamonde, A. (2012). Multilabel classifiers with a probabilistic thresholding strategy. *Pattern Recognition*, 45(2):876–883.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, 85(3):333.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2021). Classifier chains: a review and perspectives. *Journal of Artificial Intelligence Research*, 70:683–718.

Trohdis, K. (2008). Multi-label classification of music into emotions. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, pages 325–330.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2009). Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer.

Waegeman, W., Dembczyński, K., Jachnik, A., Cheng, W., and Hüllermeier, E. (2014). On the bayes-optimality of f-measure maximizers. *The Journal of Machine Learning Research*, 15(1):3333–3388.

Yang, G., Destercke, S., and Masson, M.-H. (2014). Nested dichotomies with probability sets for multi-class classification. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence (ECAI)*, pages 363–368.

Ye, N., Chai, K. M. A., Lee, W. S., and Chieu, H. L. (2012). Optimizing f-measures: a tale of two approaches. In *Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML)*, pages 1555–1562. Omnipress.

Zhang, M.-L. and Zhou, Z.-H. (2006). Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, 18(10):1338–1351.

Zhang, M.-L. and Zhou, Z.-H. (2014). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837.