Bayes Optimal Multilabel Classification via Probabilistic Classifier Chains

Krzysztof Dembczyński[1,2] DEMBCZYNSKI@INFORMATIK.UNI-MARBURG.DE
Weiwei Cheng[1] CHENG@INFORMATIK.UNI-MARBURG.DE
Eyke Hüllermeier[1] EYKE@INFORMATIK.UNI-MARBURG.DE
[1]Mathematics and Computer Science, Marburg University, Hans-Meerwein-Str., 35032 Marburg, Germany
[2]Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965 Poznań, Poland

Abstract

In the realm of multilabel classification (MLC), it has become an opinio communis that optimal predictive performance can only be achieved by learners that explicitly take label dependence into account. The goal of this paper is to elaborate on this postulate in a critical way. To this end, we formalize and analyze MLC within a probabilistic setting. Thus, it becomes possible to look at the problem from the point of view of risk minimization and Bayes optimal prediction. Moreover, inspired by our probabilistic setting, we propose a new method for MLC that generalizes and outperforms another approach, called classifier chains, that was recently introduced in the literature.

1. Introduction

In contrast to conventional (single-label) classification, the setting of *multilabel classification* (MLC) allows an instance to belong to several classes simultaneously. At first sight, MLC problems can be solved in a quite straightforward way, namely through decomposition into several binary classification problems: One binary classifier is trained for each label and used to predict whether, for a given query instance, this label is present (relevant) or not. This approach is known as *binary relevance* (BR) learning.

However, BR has been criticized for ignoring important information hidden in the label space, namely information about the interdependencies between the labels: Since the presence or absence of the different class labels has to be predicted *simultaneously*, it is arguably important to exploit these dependencies. Today, it seems to be an opinio communis that optimal predictive performance can only be achieved by methods that explicitly take label correlations into account.

Many papers provide empirical evidence for this conjecture: A new method is proposed that exploits label correlations in one way or the other. Using a set of benchmark data sets, this method is then shown to outperform others in terms of different loss functions. Without questioning the value of these contributions, one may argue that this is not sufficient to gain a deeper understanding of the MLC problem. There are several reasons for this, notably the following.

First, the notion of "label correlation" is often used in a purely intuitive manner, referring to a kind of non-independence, but without giving a precise formal definition. Likewise, MLC methods are often ad-hoc extensions of existing methods. Second, many studies report improvements *on average*, but without carefully investigating under which conditions label correlations are crucial, and when they are perhaps less important. Third, the reasons for improvements are often not carefully distinguished. As the performance of a method depends on many factors, which are hard to isolate, it is not always clear that the improvements can be fully credited to the consideration of label correlations.

The goal of this paper is to provide a formal setting that allows for a more thorough analysis of MLC in general and label dependence in particular (Section 2). To this end, we distinguish two types of label dependence, conditional and unconditional, and focus our analysis on the former. We propose a probabilistic framework that suggests to look at the problem from the point of view of risk minimization and Bayes optimal prediction. Concretely, we analyze three types of

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

loss functions and, based on the results, raise the following conjecture: While considering conditional label dependence can indeed be useful for certain loss functions, there are others that are less likely to benefit (Section 3).

A second important contribution of this paper is a new method for MLC, called *probabilistic classifier chains* (Section 4). It estimates the entire joint distribution of labels and, therefore, allows us to experimentally confirm our theoretical claims (Section 5). This method generalizes and provides a proper interpretation of the recently introduced classifier chains (Read et al., 2009). It also outperforms this algorithm, however, at the cost of an increased computational complexity.

2. Multilabel Classification

In this section, we describe the MLC problem in more detail and formalize it within a probabilistic setting. Along the way, we introduce the notation used throughout the paper.

Let $\mathcal{X}$ denote an instance space, and let $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ be a finite set of class labels. We assume that an instance $\boldsymbol{x} \in \mathcal{X}$ is (non-deterministically) associated with a subset of labels $L \in 2^{\mathcal{L}}$; this subset is often called the set of relevant labels, while the complement $\mathcal{L} \setminus L$ is considered as irrelevant for $\boldsymbol{x}$. We identify a set L of relevant labels with a binary vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$, in which $y_i = 1 \Leftrightarrow \lambda_i \in L$. By $\mathcal{Y} = \{0, 1\}^m$ we denote the set of possible labelings.

We assume observations to be generated independently and randomly according to a probability distribution $\mathbf{P}(\mathbf{X}, \mathbf{Y})$ on $\mathcal{X} \times \mathcal{Y}$, i.e., an observation $\boldsymbol{y} = (y_1, \ldots, y_m)$ is the realization of a corresponding random vector $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_m)$. We denote by $\mathbf{P}_{\boldsymbol{x}}(\mathbf{Y}) = \mathbf{P}(\mathbf{Y} \,|\, \boldsymbol{x})$ the conditional distribution of $\mathbf{Y}$ given $\mathbf{X} = \boldsymbol{x}$, and by $\mathbf{P}^{(i)}_{\boldsymbol{x}}(Y_i) = \mathbf{P}^{(i)}(Y_i \,|\, \boldsymbol{x})$ the corresponding marginal distribution of Y_i:

$$\mathbf{P}^{(i)}_{\boldsymbol{x}}(b) = \sum_{\boldsymbol{y} \in \mathcal{Y} : y_i = b} \mathbf{P}_{\boldsymbol{x}}(\boldsymbol{y}).$$

A multilabel classifier $\mathbf{h}$ is an $\mathcal{X} \to \mathcal{Y}$ mapping that assigns a (predicted) label subset to each instance $\boldsymbol{x} \in \mathcal{X}$. Thus, the output of a classifier $\mathbf{h}$ is a vector

$$\mathbf{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x})).$$

Often, MLC is treated as a ranking problem, in which the labels are sorted according to the degree of relevance. Then, the prediction takes the form of the *ranking* or *scoring function*:

$$\mathbf{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \tag{1}$$

such that the labels λ_i are simply sorted in decreasing order according to their scores $f_i(\boldsymbol{x})$.

The problem of MLC can be stated as follows: Given training data in the form of a finite set of observations $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$, drawn independently from $\mathbf{P}(\mathbf{X}, \mathbf{Y})$, the goal is to learn a classifier $\mathbf{h} : \mathcal{X} \to \mathcal{Y}$ that generalizes well beyond these observations in the sense of minimizing the expected risk with respect to a specific loss function.

Before having a more detailed look at this problem in Section 3 below, let us note that the posterior probability distributions $\mathbf{P}_{\boldsymbol{x}}(\mathbf{Y})$ provide a convenient means for analyzing label dependence. A distribution of this kind informs about the probability of each label combination as well as the marginals, like in this simple example of the case $m = 2$:

$\mathbf{P}_{\boldsymbol{x}}(\mathbf{Y})$	0	1	$\mathbf{P}^{(1)}_{\boldsymbol{x}}(1)$
0	0.4	0.0	0.4
1	0.3	0.3	0.6
$\mathbf{P}^{(2)}_{\boldsymbol{x}}(1)$	0.7	0.3	1

In a stochastic sense, the labels are not independent if the joint conditional distribution is not the product of the marginals (like in the above example):

$$\mathbf{P}_{\boldsymbol{x}}(\mathbf{Y}) \neq \prod_{i=1}^{m} \mathbf{P}^{(i)}_{\boldsymbol{x}}(Y_i), \tag{2}$$

and the degree of dependence could in principle be quantified in terms of measures like cross entropy or KL divergence.

More specifically, we shall speak of *conditional* dependence in the case of (2), since the probability of $\mathbf{Y}$ is conditioned on the instance $\boldsymbol{x}$. This type of (in)dependence can be distinguished from *unconditional* (in)dependence, which looks at the unconditional probabilities of labels, i.e., $\mathbf{P}(\mathbf{Y})$ and $\mathbf{P}^{(i)}(Y_i)$ obtained, respectively by integrating $\mathbf{P}_{\boldsymbol{x}}(\mathbf{Y})$ and $\mathbf{P}^{(i)}_{\boldsymbol{x}}(Y_i)$ over all $\boldsymbol{x}$. One readily verifies that conditional does not imply unconditional dependence nor the other way around.

Upon closer examination, it becomes clear that several MLC methods, including those based on the idea of stacking (Godbole & Sarawagi, 2004; Cheng & Hüllermeier, 2009), seek to exploit unconditional label dependence, and so do related methods from other fields, like multivariate regression in statistics (Breiman & Friedman, 1997). As will become clear later on, however, Bayes optimal prediction may require the consideration of conditional dependence. In this paper, we shall therefore focus on this type of dependence in the first place.

3. Risk Minimization

In this section, we address the issue of optimal decision making in light of different loss functions. In general, we would like to find a risk-minimizing model $\mathbf{h}^*$, i.e., a model that minimizes the expected loss over the joint distribution $\mathbf{P}(\mathbf{X}, \mathbf{Y})$:

$$R(\mathbf{h}) = \mathbb{E}_{\mathbf{XY}} L(\mathbf{Y}, \mathbf{h}(\mathbf{X})),$$

where $L(\cdot)$ is a loss function on multilabel predictions. This model is given by

$$\mathbf{h}^* = \arg\min_{\mathbf{h}} \mathbb{E}_{\mathbf{XY}} L(\mathbf{Y}, \mathbf{h}(\mathbf{X})) \tag{3}$$

and determined in a pointwise way by the *Bayes optimal decisions*

$$\mathbf{h}^*(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}} \mathbb{E}_{\mathbf{Y}|\mathbf{X}} L(\mathbf{Y}, \boldsymbol{y}). \tag{4}$$

In the following, we consider the problem of risk minimization for three different types of loss functions that are frequently used in the context of MLC and to some extent representative of losses used in this field.

3.1. Hamming Loss

The performance in MLC is perhaps most frequently reported in terms of the Hamming loss, which is defined as the number (or, in its normalized version, the fraction) of labels whose relevance is incorrectly predicted:[1]

$$L_H(\boldsymbol{y}, \mathbf{h}(\boldsymbol{x})) = \sum_{i=1}^{m} [\![y_i \neq h_i(\boldsymbol{x})]\!]. \tag{5}$$

For the Hamming loss (5), it is easy to see that the risk minimizer (4) is obtained by

$$h_i^*(\boldsymbol{x}) = \arg\max_{b \in \{0,1\}} \mathbf{P}_{\boldsymbol{x}}^{(i)}(b). \tag{6}$$

In fact, the Hamming loss is a sum of the conventional 0/1 loss functions widely used in binary classification, so the form of the risk minimizer is not surprising.

3.2. Rank Loss

Another loss function that is commonly used in MLC is the rank loss. Instead of comparing two label subsets, this loss function compares the true label subset with a predicted ranking (total order) of labels, as represented by the ranking function (1). In this ranking, all relevant labels ideally precede all irrelevant ones, and the rank loss simply counts the number of label pairs violating this condition:

$$L_r(\boldsymbol{y}, \mathbf{f}(\boldsymbol{x})) = \sum_{(i,j):y_i > y_j} \left([\![f_i < f_j]\!] + \frac{1}{2} [\![f_i = f_j]\!] \right). \tag{7}$$

In passing, we note that there is also a normalized variant of the rank loss, in which this number is divided by the maximum number of possible mistakes on $\boldsymbol{y}$, i.e., by the number of summands in (7); this number is given by $r(m-r)/2$, with $r = y_1 + \ldots + y_m$ the number of relevant labels.

To minimize (7), it is enough to sort the labels by their probability of relevance. Formally, we can show the following result.

Theorem 3.1. *A ranking function that sorts the labels according to their probability of relevance, i.e., using the scoring function* $\mathbf{f}(\cdot)$ *with*

$$f_i(\boldsymbol{x}) \;=\; \mathbf{P}_{\boldsymbol{x}}^{(i)}(1) \;, \tag{8}$$

minimizes the expected rank loss (7).

Proof. The risk of a scoring vector $\mathbf{f} = \mathbf{f}(\boldsymbol{x})$ can be written as

$$\mathbb{E}_{\mathbf{Y}\,|\,\mathbf{X}} L_r(\mathbf{Y}, \mathbf{f}) = \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbf{P}_{\boldsymbol{x}}(\boldsymbol{y}) L_r(\boldsymbol{y}, \mathbf{f}) =$$

$$= \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbf{P}_{\boldsymbol{x}}(\boldsymbol{y}) \sum_{y_i > y_j} \left([\![f_i < f_j]\!] + \frac{1}{2} [\![f_i = f_j]\!] \right).$$

The two sums can be swapped, and doing so yields the expression

$$\sum_{1 \leq i,j \leq m} \sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbf{P}_{\boldsymbol{x}}(\boldsymbol{y}) [\![y_i > y_j]\!] \left([\![f_i < f_j]\!] + \frac{1}{2} [\![f_i = f_j]\!] \right)$$

which in turn can be written as

$$\sum_{1 \leq i < j \leq m} g(i,j) + g(j,i)$$

with

$$g(i,j) = \mathbf{P}_{\boldsymbol{x}}(y_i > y_j) \left([\![f_i < f_j]\!] + \frac{1}{2} [\![f_i = f_j]\!] \right).$$

For each pair of labels y_i, y_j, the sum $g(i,j) + g(j,i)$ is obviously minimized by choosing the scores f_i, f_j such that $f_i \leq f_j$ if and only if $\mathbf{P}_{\boldsymbol{x}}(y_i > y_j) \leq \mathbf{P}_{\boldsymbol{x}}(y_j > y_i)$, and since $\mathbf{P}_{\boldsymbol{x}}(y_i > y_j) - \mathbf{P}_{\boldsymbol{x}}(y_j > y_i) = \mathbf{P}_{\boldsymbol{x}}^{(i)}(1) - \mathbf{P}_{\boldsymbol{x}}^{(j)}(1)$, the condition on the right-hand side is equivalent to $\mathbf{P}_{\boldsymbol{x}}^{(i)}(1) \leq \mathbf{P}_{\boldsymbol{x}}^{(j)}(1)$. Consequently, the scores (8) minimize the sums $g(i,j) + g(j,i)$ simultaneously for all label pairs and, therefore, minimize risk. □

[1] For a predicate P, the expression $[\![P]\!]$ evaluates to 1 if P is true and to 0 if P is false.

3.3. Subset Zero-One Loss

Finally, it is of course also possible to generalize the well-known 0/1 loss from the conventional to the multilabel setting:

$$L_s(\boldsymbol{y}, \mathbf{h}(\boldsymbol{x})) = [\![\boldsymbol{y} \neq \mathbf{h}(\boldsymbol{x})]\!] \ . \tag{9}$$

This loss function is referred to as *subset zero-one* loss. Admittedly, it may appear overly stringent, especially in the case of many labels. Moreover, since making a mistake on a single label is punished as hardly as a mistake on all labels, it does not discriminate well between "almost correct" and completely wrong predictions. Still, this measure is obviously interesting with regard to dependence between labels.

The Bayes prediction for (9) is rather straight-forward. As for any other 0/1 loss, it simply consists of predicting the mode of the distribution:

$$\mathbf{h}^*(\boldsymbol{x}) = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \mathbf{P}_{\boldsymbol{x}}(\boldsymbol{y}) \ . \tag{10}$$

3.4. Consequences and Conjectures

As one of the most important consequences of the above results we note that, according to (6) and (7), a risk-minimizing prediction for the Hamming and the rank loss can be obtained from the marginal distributions $\mathbf{P}_{\boldsymbol{x}}^{(i)}(Y_i)$ $(i = 1, \ldots, m)$ alone. In other words, it is not necessary to know the joint label distribution $\mathbf{P}_{\boldsymbol{x}}(\mathbf{Y})$ on $\mathcal{Y}$. As opposed to this, (10) shows that the entire distribution of $\mathbf{Y}$ given $\boldsymbol{x}$ (or at least the mode of this distribution) is needed to minimize the subset zero-one loss. Let us remark, however, that for conditionally independent labels, the risk minimizers for the Hamming and the subset 0/1 loss are exactly the same. The same holds if the probability of the mode is greater or equal 0.5.

Now, since marginal distributions can in principle be estimated independently of each other, without taking conditional dependencies into consideration, we claim that, while measures like (9) may indeed benefit from MLC methods that are able to exploit conditional dependencies, the gain, if any, will be much smaller for measures like Hamming and rank loss.

In Section 5, we shall test this conjecture in an empirical way. Before we can do so, we need a method that is able to estimate Bayes optimal predictions. A method of this kind, that is, a learning algorithm producing a model that takes an instance $\boldsymbol{x}$ as input and produces the distribution $\mathbf{P}_{\boldsymbol{x}}(\mathbf{Y})$ as output, is proposed in the next section.

4. Probabilistic Classifier Chains

Given a query instance $\boldsymbol{x}$, the (conditional) probability of each label combination $\boldsymbol{y} = (y_1, \ldots, y_m) \in \mathcal{Y}$ can be computed using the product rule of probability:

$$\mathbf{P}_{\boldsymbol{x}}(\boldsymbol{y}) = \mathbf{P}_{\boldsymbol{x}}(y_1) \cdot \prod_{i=2}^{m} \mathbf{P}_{\boldsymbol{x}}(y_i \,|\, y_1, \ldots, y_{i-1}) \tag{11}$$

Thus, to estimate the joint distribution of labels, one possibility is to learn m functions $f_i(\cdot)$ on an augmented input space $\mathcal{X} \times \{0,1\}^{i-1}$, taking $y_1, \ldots, y_{i-1}$ as additional attributes:

$$\begin{array}{rcl} f_i : \mathcal{X} \times \{0,1\}^{i-1} & \to & [0,1] \\ (\boldsymbol{x}, y_1, \ldots, y_{i-1}) & \mapsto & \mathbf{P}(y_i = 1 \,|\, \boldsymbol{x}, y_1, \ldots, y_{i-1}) \end{array}$$

We assume here that the function $f_i(\cdot)$ can be interpreted as a *probabilistic* classifier whose prediction is the probability that $y_i = 1$, or at least a reasonable approximation thereof. Thus, (11) becomes

$$\mathbf{P}_{\boldsymbol{x}}(\boldsymbol{y}) = f_1(\boldsymbol{x}) \cdot \prod_{i=2}^{m} f_i(\boldsymbol{x}, y_1, \ldots, y_{i-1}) \tag{12}$$

Given $\mathbf{P}_{\boldsymbol{x}}$ (and a loss function $L(\cdot)$ to be minimized), an optimal prediction (4) can then be derived in an explicit way. This approach will subsequently be referred to as the *probabilistic classifier chain* (PCC).

4.1. The Original Classifier Chain

Our method is inspired by the *classifier chain* (CC) that was recently proposed by Read et al. (2009) as a meta-technique for MLC. More specifically, what they propose is the above idea of "chaining" classifiers, albeit without any connection to probability theory. Their approach works as follows: One classifier h_i is trained for each label similarly to our scoring function f_i. Given a new instance $\boldsymbol{x}$ to be classified, the model h_1 predicts y_1, i.e., the relevance of λ_1 for $\boldsymbol{x}$, as usual. Then, h_2 predicts the relevance of λ_2, taking $\boldsymbol{x}$ *plus the predicted value* $y_1 \in \{0,1\}$ as an input. Proceeding in this way, h_i predicts y_i using $y_1, \ldots, y_{i-1}$ as additional input information.

Interestingly, the original classifier chain can be seen as a deterministic approximation of (12), in the sense of using $\{0,1\}$-valued probabilities. In fact, CC is recovered from (12) in the special case where the $f_i(\cdot)$ output either 0 or 1. This results in the estimation

$$\mathbf{P}_{\boldsymbol{x}}(\boldsymbol{y}) = [\![\boldsymbol{y} = \boldsymbol{y}_{CC}]\!], \tag{13}$$

where $\boldsymbol{y}_{CC}$ is the label combination predicted by the classifier chain, which pretends certainty about the estimation $\boldsymbol{y}_{CC}$.

Needless to say, (13) will normally be a poor estimation of the true distribution $\mathbf{P}_{\boldsymbol{x}}(\mathbf{Y})$. In fact, considering the idea of chaining classifiers as searching a path in a binary tree whose leaf nodes are associated with a labeling $\boldsymbol{y} \in \mathcal{Y}$, and with 0/1 branches for y_i on level i, CC just follows a single path in this tree in a greedy manner. It can be seen as a "mode seeker", hoping to find the labeling $\boldsymbol{y}^*$ with highest probability. Due to its greedy nature, however, the mode will not always be reached.

For example, suppose that the base classifiers produce exact probability estimates, and that CC turns a probability estimate p into the prediction $[\![p > 0.5]\!]$. Then, it is easy to show that $\boldsymbol{y}_{CC} = \boldsymbol{y}^*$ if $\mathbf{P}_{\boldsymbol{x}}(\boldsymbol{y}^*) > 0.5$. If the probability of the mode is smaller than 1/2, however, CC may fail. As a small illustration, consider the case $m = 3$ and suppose that $\mathbf{P}_{\boldsymbol{x}}(0,0,1) = 0.4$, $\mathbf{P}_{\boldsymbol{x}}(1,0,1) = 0.25$ and $\mathbf{P}_{\boldsymbol{x}}(1,1,0) = 0.35$. In this case, CC will already start incorrectly, namely with $y_1 = 1$, and eventually produce the suboptimal prediction $\boldsymbol{y} = (1,1,0)$.

4.2. Complexity

The hope that PCC will produce better estimates is clearly justified in light of these observations. Of course, the price to pay is a much higher complexity. In fact, while CC searches only a single path in the aforementioned binary tree, PCC has to look at each of the 2^m paths. This limits the applicability of the method to data sets with a small to moderate number of labels, say, not more than about 15.

First, however, apart from the fact that complexity is not our main concern here, one may argue that several other methods suffer from the same problem, including the label power-set approach (Tsoumakas & Katakis, 2007), as also graphical models estimating the joint distribution (Ghamrawi & McCallum, 2005). Second, there are possibilities to develop approximate inference schemes that trade off accuracy against efficiency in a reasonable way, lying somehow in-between the exact inference (12) and the extremely crude approximation (13). This can be done in different ways, for example by pruning single labels (with provably low probability of relevance), or by ignoring label combinations with low probability (to minimize the subset zero-one loss, only the most probable label combination is needed). We can also try to factorize the high-dimensional joint distribution into several lower-dimensional distributions, exploiting label independence whenever possible.

4.3. Chain Ensembles

Theoretically, the result of the product rule does not depend on the order of the variables. Practically, two different classifier chains will produce different results, simply because they involve different classifiers learned on different training sets. To reduce the influence of the label order, Read et al. (2009) propose to average the multilabel predictions of CC over a (randomly chosen) set of permutations. Thus, the labels $\lambda_1, \ldots, \lambda_m$ are first re-ordered by a permutation π of $\{1, \ldots, m\}$, which moves the label λ_i from position i to position $\pi(i)$, and CC is then applied as usual. This extension is called the *ensembled classifier chain* (ECC).

Of course, the same idea can be applied to our probabilistic variant. In this case, it is natural to average over the predicted distributions $\mathbf{P}_{\boldsymbol{x}}(\mathbf{Y})$ directly. We call this approach the *ensembled probabilistic classifier chain* (EPCC).

5. Experimental Results

In this section, we describe experimental studies on two artificial data sets and twelve benchmark data sets that we performed in order to verify our theoretical conjectures. The experiments are specifically designed to show that the performance of a given classifier strongly depends on the measure used. We have considered the three measures discussed above: Hamming loss, rank loss, and the subset 0/1 loss.

We have performed 3-fold cross-validation for each data set except the large benchmark data sets (more than 10000 instances) for which we used the 66% split. The results are reported as an average of these measures over test instances. Moreover, for each data set we provide a ranking of the algorithms. For the benchmark data, the two-step procedure suggested by Demšar (2006) was used to test for statistically significant differences between the algorithms.

We have considered five classifiers: binary relevance (BR), classifier chains (CC), probabilistic classifier chains (PCC), ensembled classifier chains (ECC), ensembled probabilistic classifier chains (EPCC). BR is the simplest approach that treats labels independently. According to our theoretical considerations, this approach is well-suited for the Hamming and the rank loss. Let us remind that CC is well-suited for the subset 0/1 loss, since it approximates the mode of the joint distribution. ECC averages over several CC predictions, however, it is rather difficult to say, what this approach tends to estimate. The probabilistic versions of these algorithms, PCC and EPCC, are well-suited for all measures, since corresponding risk minimizers

can be computed from the joint distribution estimated by these methods.

We tried to eliminate as many as possible additional effects that may bias the results. Since all algorithms are meta-learners, we used logistic regression without regularization as a base learner for all of them. This choice is also motivated by the fact that logistic regression provides (conditional) probability estimates as predictions. CC and ECC are implemented according to (Read et al., 2009) with minor exceptions. We have not performed additional sampling from the data set for training the ensemble members in ECC, and we did not tune the threshold for predicting class labels; we simply set it to 0.5 (as in all algorithms). Permutations of labels in all chain-based algorithms have been generated at random. Let us underline that learning for (E)CC and (E)PCC is the same, and the difference concerns the prediction phase only. The ensemble size in ECC and EPCC was set to 10.

Let us also notice that the results highly depend on data sets used. For data sets without dependencies among the labels or with high probability of the joint modes (≥ 0.5), the risk minimizers for the Hamming and the subset 0/1 loss coincide. Of course, it is hard to estimate the nature of a data set in advance, since even a high correlation between labels in the data set does not mean that the joint distribution for given $\boldsymbol{x}$ shares these dependencies and vice versa.

5.1. Artificial Data

We have used two artificial data sets with three labels. The first one is a collection of independent problems. In the second one, the labels are strongly dependent. The data models used for generating these data sets are supposed to be as simple as possible. For each data set, we generated 10000 instances.

The independent data set was generated by uniformly drawing instances from the square $\boldsymbol{x} \in [-0.5, 0.5]^2$. The label distribution is given by the product of the marginal distributions defined by $\mathbf{P}_{\boldsymbol{x}}(y_i) = 1/(1+\exp(-f_i(\boldsymbol{x}))$, where the f_i are linear functions: $f_1(\boldsymbol{x}) = x_1 + x_2$, $f_2(\boldsymbol{x}) = -x_1 + x_2$, $f_3(\boldsymbol{x}) = x_1 - x_2$. The cardinality of labels (the average number of relevant labels for an instance) is 1.503.

The dependent data set was generated by drawing the instances from a univariate uniform distribution $\boldsymbol{x} \in [-0.5, 0.5]$. The label distribution is given by the product rule[2]: $\mathbf{P}_{\boldsymbol{x}}(\mathbf{Y}) = \mathbf{P}_{\boldsymbol{x}}(y_1)\mathbf{P}_{\boldsymbol{x}}(y_2 \,|\, y_1)\mathbf{P}_{\boldsymbol{x}}(y_3 \,|\, y_1, y_2)$, where the probabilities are modeled by linear functions in a similar way as before: $f_1(x) = x$, $f_2(y_1, x) = -x - 2y_1 + 1$, $f_3(y_2, y_1, x) = x + 12y_1 - 2y_2 - 11$. The cardinality of labels for this data set is 1.314.

[2]We have experimented with several models of dependence and obtained similar results; the product rule model is presented, since it is prominently discussed in the paper.

Table 1. Results on two artificial data sets: independent model (top) and dependent model (down).

classifier	Hamming loss	rank loss	subset 0/1 loss
BR	0.4178(2)	0.5527(1)	0.8108(3)
CC	0.4189(4.5)	0.5934(5)	0.8124(5)
PCC	0.4178(2)	0.5528(2.5)	0.8088(1.5)
ECC	0.4189(4.5)	0.5907(4)	0.8120(4)
EPCC	0.4178(2)	0.5528(2.5)	0.8088(1.5)
B-O	0.4179	0.5532	0.8088
BR	0.3921(3)	0.5675(2)	0.7374(5)
CC	0.4308(4)	0.6930(4)	0.6100(3)
PCC	0.3920(1.5)	0.5676(3)	0.6052(2)
ECC	0.4320(5)	0.6954(5)	0.6112(4)
EPCC	0.3920(1.5)	0.5674(1)	0.6051(1)
B-O	0.3920	0.5671	0.6057

The results are shown in Table 5.1. In the case of the independent data we can see that, in agreement with our theoretical considerations, there is almost no difference between the algorithms with respect to the Hamming and the subset 0/1 loss. The poor performance of CC and ECC on rank loss follows from the fact that these methods do not estimate the marginals. For the dependent data, we see that PCC and EPCC adapt to the loss function. Moreover, as expected, BR performs well for the Hamming and the rank loss. CC and ECC are better than BR with respect to the subset 0/1 loss, which is in agreement with our theoretical results. However, we observe that both algorithms approximate the optimal prediction in a suboptimal way and are outperformed by PCC and EPCC. It seems that ensembling the CC classifiers does not improve performance.

Since both models are known, we have also computed the Bayes-optimal predictions (denoted by B-O). By comparing the performance to B-O, one can measure the regret, that is, the drop in performance caused by not being well-adapted to the loss (since B-O predictions are computed on a finite sample, some classifiers may obtain slightly better results).

5.2. Benchmark Data

The second part of the experiment was performed on a relatively large collection of 12 multilabel classification data sets[3]. A summary of the data sets and

[3]Taken from mlkd.csd.auth.gr/multilabel.html and www.cs.waikato.ac.nz/~jmr30/#datasets

their properties are given in Table 2. Since PCC and EPCC are computationally complex, we have limited the number of labels to 10. For each data set, we kept the most frequent labels (and subsequently removed all instances having only relevant or only irrelevant labels). The *reuters* data set has been preprocessed as in (Cheng & Hüllermeier, 2009).

The results are summarized in Table 3. We conducted the Friedman test based on the average ranks in order to verify whether the difference between algorithms are statistically significant. For each loss function, the null hypothesis was rejected at a significance level of 5%. According to the post-hoc analysis based on Nemenyi statistics, the significant difference in average ranks of the algorithms is 1.84.

The results are in agreement with our theoretical conjectures and the previous experiment on artificial data. The EPCC algorithm performs best, as it can be tailored for all loss functions and benefits from averaging over the ensemble. It seems that for those benchmark data sets that are much larger in the feature and label space, ensembling produces a significant improvement: Both EPCC and ECC outperform their respective regular variant (for the Hamming and the rank loss, the difference is significant).

We note, however, that the comparison between ensemble and non-ensemble methods is not completely fair. This is why we claim that BR performs well with respect to the Hamming and the rank loss. Using a more powerful classifier in BR, we even expect the difference to EPCC to decrease further. The worst in this case is the CC algorithm, as it is not tailored for these losses. In the case of the subset 0/1 loss, BR is obviously the worst, and CC obtains better results. It is interesting that ECC has the second lowest average rank for the Hamming and the rank loss, which may suggest that the averaging used in this classifier brings the prediction closer to the marginals. However, this method has also obtained the second lowest average rank for the subset 0/1 loss. The interpretation of this result is difficult, since we do not know the real dependencies between labels. It seems that some of the data sets do not contain strong dependencies or the probability of joint modes is high (≥ 0.5), which could partly explain the overall good performance of ECC.

6. Conclusions

We proposed a probabilistic framework of multilabel classification and analyzed the risk minimizers of three common loss functions. Even though most of the theoretical results are quite obvious, they provide some important insights into the nature of MLC. In particular, they show that the main concern of recent contributions to MLC, namely the exploitation of label dependence, should be considered with diligence: First, one has to distinguish two types of dependence, conditional and unconditional, and second, the latter is strongly related to the loss function to be minimized.

A second contribution is a new MLC method, probabilistic classifier chains, that extends the recently introduced CC classifier. This algorithm is able to estimate the entire joint distribution of the labels and, therefore, can be tailored to any loss function. Its drawback is the computational complexity at prediction time. We therefore plan to develop approximate inference schemes that trade off complexity against accuracy of probability estimation in an optimal way.

Our analysis has also shed light on the CC classifier itself that was lacking a sound theoretical motivation so far. Here, we have shown that it can be seen as a greedy approximation of the most probable label combination. This interpretation enables a formal analysis of the algorithm.

Acknowledgments

We thank Willem Waegeman for interesting discussions and useful suggestions. This work has been supported by the Germany Research Foundation (DFG).

References

Breiman, L. and Friedman, J. Predicting multivariate responses in multiple linear regression. *J R Stat Soc B*, 69:3–54, 1997.

Cheng, W. and Hüllermeier, E. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76:211–225, 2009.

Demšar, J. Statistical comparisons of classifiers over multiple data sets. *JMRL*, 7:1–30, 2006.

Ghamrawi, N. and McCallum, A. Collective multi-label classification. In *CIKM '05*, pp. 195–200, 2005.

Godbole, S. and Sarawagi, S. Discriminative methods for multi-labeled classification. In *PAKDD 2004*, pp. 22–30, 2004.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. Classifier chains for multi-label classification. In *ECML/PKDD 2009*, pp. 254–269, 2009.

Tsoumakas, G. and Katakis, I. Multi label classification: An overview. *Int J Data Warehouse and Mining*, 3:1–13, 2007.

Table 2. Data sets used in the experiment.

data set	# inst.	# attr.	# labels	cardinality	data set	# inst.	# attr.	# labels	cardinality
image	2000	135	5	1.236	enron-10	1677	1001	10	2.513
scene	2407	294	6	1.074	medical-10	785	1449	10	1.195
emotions	593	72	6	1.868	slashdot-10	3279	1079	10	1.134
reuters	7119	243	7	1.241	ohsumed-10	11934	1002	10	1.461
yeast-10	2389	103	10	3.971	tmc2007-500-10	27409	500	10	1.979
mediamill-10	41583	120	10	3.175	imdb-10	86290	1001	10	1.623

Table 3. Results on 12 benchmark data sets.

	BR	CC	PCC	ECC	EPCC
			Hamming loss		
image	0.2063 (2)	0.2279 (5)	0.2098 (3.5)	0.2098 (3.5)	0.2033 (1)
scene	0.1644 (3)	0.1780 (4.5)	0.1780 (4.5)	0.1503 (2)	0.1498 (1)
emotions	0.2395 (2)	0.2448 (5)	0.2417 (3)	0.2428 (4)	0.2372 (1)
reuters	0.0546 (3)	0.0593 (5)	0.0583 (4)	0.0520 (2)	0.0514 (1)
yeast-10	0.2602 (3)	0.2736 (5)	0.2593 (2)	0.2651 (4)	0.2583 (1)
mediamill-10	0.1628 (3)	0.1695 (5)	0.1621 (2)	0.1646 (4)	0.1619 (1)
enron-10	0.3157 (3)	0.3196 (5)	0.3193 (4)	0.3110 (1)	0.3111 (2)
medical-10	0.0648 (5)	0.0641 (4)	0.0640 (3)	0.0550 (2)	0.0547 (1)
slashdot-10	0.2230 (5)	0.1888 (4)	0.1887 (3)	0.1683 (1.5)	0.1683 (1.5)
ohsumed-10	0.1386 (3)	0.1398 (5)	0.1387 (4)	0.1356 (2)	0.1344 (1)
tmc-500-10	0.1038 (4)	0.1059 (5)	0.1029 (2)	0.1035 (3)	0.1026 (1)
imdb-10	0.1632 (3)	0.1916 (5)	0.1629 (1)	0.1806 (4)	0.1630 (2)
Ave. Rank	3.25	4.792	3	2.75	1.208
			Rank loss		
image	0.9510 (2)	1.0910 (5)	0.9869 (4)	0.9525 (3)	0.9439 (1)
scene	1.0262 (3)	1.1554 (5)	1.0841 (4)	0.8884 (2)	0.8855 (1)
emotions	1.4367 (3)	1.5816 (5)	1.5445 (4)	1.4332 (1)	1.4349 (2)
reuters	0.2598 (1)	0.3494 (5)	0.3144 (4)	0.2748 (3)	0.2698 (2)
yeast-10	4.4562 (1)	5.3347 (5)	4.5115 (3)	4.6346 (4)	4.4617 (2)
mediamill-10	2.1063 (3)	2.4565 (5)	2.0755 (2)	2.1631 (4)	2.0720 (1)
enron-10	6.0247 (3)	6.1410 (4)	6.1655 (5)	5.9213 (2)	5.8915 (1)
medical-10	0.7638 (5)	0.7584 (4)	0.7240 (3)	0.5096 (2)	0.4982 (1)
slashdot-10	3.0933 (4)	3.1232 (5)	3.0461 (3)	2.7585 (2)	2.7219 (1)
ohsumed-10	2.1059 (3)	2.1489 (5)	2.1067 (4)	1.9556 (2)	1.9323 (1)
tmc-500-10	0.9518 (4)	0.9742 (5)	0.9471 (3)	0.9462 (2)	0.9366 (1)
imdb-10	3.4139 (1)	3.7588 (5)	3.4424 (3)	3.4514 (4)	3.4260 (2)
Ave. Rank	2.75	4.833	3.5	2.583	1.333
			Subset 0/1 loss		
image	0.6900 (5)	0.6260 (4)	0.6095 (2.5)	0.6095 (2.5)	0.5890 (1)
scene	0.6448 (5)	0.5954 (4)	0.5949 (3)	0.5505 (2)	0.5123 (1)
emotions	0.8178 (5)	0.7757 (2)	0.7723 (1)	0.7976 (4)	0.7790 (3)
reuters	0.2889 (5)	0.2860 (4)	0.2843 (3)	0.2653 (2)	0.2583 (1)
yeast-10	0.8393 (5)	0.7861 (3)	0.7761 (2)	0.8049 (4)	0.7710 (1)
mediamill-10	0.7965 (5)	0.7690 (3)	0.7517 (2)	0.7698 (4)	0.7504 (1)
enron-10	0.9350 (3)	0.9416 (4.5)	0.9416 (4.5)	0.9320 (2)	0.9302 (1)
medical-10	0.4497 (5)	0.4370 (4)	0.4357 (3)	0.3885 (1)	0.3898 (2)
slashdot-10	0.8866 (5)	0.8588 (4)	0.8582 (3)	0.8112 (1)	0.8286 (2)
ohsumed-10	0.7650 (5)	0.7568 (3)	0.7575 (4)	0.7459 (1)	0.7489 (2)
tmc-500-10	0.6429 (5)	0.6364 (4)	0.6268 (2)	0.6294 (3)	0.6242 (1)
imdb-10	0.9463 (5)	0.8391 (3)	0.8239 (1)	0.8411 (4)	0.8259 (2)
Ave. Rank	4.833	3.542	2.583	2.542	1.5