On Label Dependence in Multi-Label Classification

Krzysztof Dembczynski[1,2] DEMBCZYNSKI@INFORMATIK.UNI-MARBURG.DE
Willem Waegeman[3] WILLEM.WAEGEMAN@UGENT.BE
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., 35039 Marburg, Germany
[2]Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965 Poznań, Poland
[3]Department of Applied Mathematics, Biometrics and Process Control, Ghent University, Coupure links 653, 9000 Ghent, Belgium

Abstract

The aim of this paper is to elaborate on the important issue of label dependence in multi-label classification (MLC). Looking at the problem from a statistical perspective, we claim that two different types of label dependence should be distinguished, namely conditional and unconditional. We formally explain the differences and connections between both types of dependence and illustrate them by means of simple examples. Moreover, we given an overview of state-of-the-art algorithms for MLC and categorize them according to the type of label dependence they seek to capture.

1. Introduction

In current research on multi-label classification (MLC), it seems to be an opinio communis that optimal predictive performance can only be achieved by methods that explicitly account for possible dependencies between class labels. Indeed, there is an increasing number of papers providing evidence for this conjecture, mostly by virtue of empirical studies. Often, a new approach to exploiting label dependence is proposed, and the corresponding method is shown to outperform others in terms of different loss functions. Without questioning the potential benefit of exploiting label dependencies in general, we argue that studies of this kind do often fall short at deepening the understanding of the MLC problem. There are several reasons for this, notably the following.

Appearing in *Working Notes of the 2nd International Workshop on Learning from Multi-Label Data*, Haifa, Israel, 2010.

First, the notion of label dependence or "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 for multi-class classification. Second, many studies report improvements *on average*, but without carefully investigating the conditions under which label correlations are useful. 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 aim of this paper is to elaborate on the issue of label dependence in more detail, thereby helping to gain a better understanding of MLC in general. In particular, we make the point that two different types of dependence should be distinguished when talking about label dependence in MLC. These two types will be referred to as *conditional* and *unconditional* label dependence, respectively. While the latter captures dependencies between labels conditional to a specific instance, the former is a global type of dependence independent of any concrete observation.

We formally explain the differences and connections between both types of dependence and illustrate them by means of simple examples. As a conclusion, we will claim that both types of dependence can be useful to improve the performance of multi-label classifiers. However, it will also become clear that substantially different algorithms are needed to exploit conditional and unconditional dependence, respectively. In this regard, we also give an overview of state-of-the-art algorithms for MLC and categorize them according to the type of label dependence they seek to capture.

2. Multi-Label 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 multi-label 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

$$\mathbf{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x})). \tag{1}$$

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 a *ranking* or *scoring function*:

$$\mathbf{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \tag{2}$$

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 risk with respect to a specific loss function. Commonly used loss function, also to be considered later on in this paper, include the subset 0/1 loss, the Hamming loss, and the rank loss.

The subset 0/1 loss generalizes the well-known 0/1 loss from the single-label to the multi-label setting:[1]

$$L_s(\boldsymbol{y}, \mathbf{h}(\boldsymbol{x})) = [\![\boldsymbol{y} \neq \mathbf{h}(\boldsymbol{x})]\!] \ . \tag{3}$$

The Hamming loss can be seen as another generalization that averages over the standard 0/1 losses of the different labels:

$$L_H(\boldsymbol{y}, \mathbf{h}(\boldsymbol{x})) = \frac{1}{m} \sum_{i=1}^{m} [\![y_i \neq h_i(\boldsymbol{x})]\!]. \tag{4}$$

Instead of comparing two label subsets, the rank loss compares the true label subset with a predicted ranking (total order) of labels, as represented by the ranking function (2). More specifically, it counts the number of cases in which an irrelevant label precedes a relevant label:

$$L_r(\boldsymbol{y}, \mathbf{f}(\boldsymbol{x})) = \sum_{(i,j) : y_i > y_j} \left([\![f_i < f_j]\!] + \frac{1}{2} [\![f_i = f_j]\!] \right). \tag{5}$$

3. Label Dependence

As mentioned previously, we distinguish two types of label dependence in MLC, namely conditional and unconditional dependence. We start with a formal definition of the latter.

Definition 3.1. *A vector of labels*

$$\mathbf{Y} = (Y_1, Y_2, \ldots, Y_m) \tag{6}$$

is called unconditionally m-independent if

$$\mathbf{p}(\mathbf{Y}) = \prod_{i=1}^{m} \mathbf{p}^{(i)}(Y_i) \,. \tag{7}$$

Remark that m refers to the number of random variables, i.e., labels in our context, and that we do not, unlike stochastic (mutual) independence, require (7) to hold for all subsets of variables, too. In the following, we shall simply speak of independence, always referring to m-independence unless otherwise stated.

Conditional dependence captures the dependence of the labels given a specific instance $\boldsymbol{x} \in \mathcal{X}$.

Definition 3.2. *A vector of labels (6) is called conditionally m-independent given $\boldsymbol{x}$ if*

$$\mathbf{p}_{\boldsymbol{x}}(\mathbf{Y}) = \prod_{i=1}^{m} \mathbf{p}^{(i)}_{\boldsymbol{x}}(Y_i)$$

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

Recall that the joint distribution of a random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_m)$ can be expressed by the product rule of probability:

$$\mathbf{p}(\boldsymbol{Y}) = \mathbf{p}(Y_1) \prod_{i=2}^{m} \mathbf{p}(Y_i | Y_1, \ldots, Y_{i-1}). \tag{8}$$

If $Y_1, \ldots, Y_m$ are m-independent, then (8) simplifies to (7). The same remark applies to the joint conditional probability.

The above two types of dependence may look very similar, since they only differ in the use of unconditional and conditional probability measures. Moreover, we have a strong connection between unconditional and conditional dependence, since

$$\mathbf{p}(\mathbf{Y}) = \int_{\mathcal{X}} \mathbf{p}_{\boldsymbol{x}}(\mathbf{Y}) \, d\mathbf{P}(\boldsymbol{x}) \,. \tag{9}$$

Roughly speaking, unconditional dependence is a kind of "expected dependence", averaged over all instances. Despite this close connection, one can easily construct examples showing that conditional dependence does not imply unconditional dependence nor the other way around.

Example 3.1. *Consider a problem with two labels y_1 and y_2, both being independently generated through the same logistic model $\mathbf{p}^{(i)}_{\boldsymbol{x}}(1) = (1 + \exp(-\phi f(\boldsymbol{x})))^{-1}$, where ϕ corresponds to the Bayes rate. Thus, by definition, the two labels are conditionally independent, having joint distribution $\mathbf{p}_{\boldsymbol{x}}(\mathbf{Y}) = \mathbf{p}_{\boldsymbol{x}}(Y_1) \times \mathbf{p}_{\boldsymbol{x}}(Y_2)$ given $\boldsymbol{x}$. However, depending on the value of ϕ, we will have a stronger or weaker unconditional dependence. For $\phi \to \infty$ (Bayes rate tends to 0), the unconditional dependence increases toward an almost deterministic one ($y_1 = y_2$).*

The next example shows that conditional dependence does not imply unconditional dependence.

Example 3.2. *Consider a problem in which two labels y_1 and y_2 are to be predicted by using a single binary feature x_1. Let us assume that the joint distribution $\mathbf{p}(X_1, Y_1, Y_2)$ on $\mathcal{X} \times \mathcal{Y}$ is given as in the following table:*

x_1	y_1	y_2	$\mathbf{p}$	x_1	y_1	y_2	$\mathbf{p}$
0	*0*	*0*	*0.25*	*1*	*0*	*0*	*0*
0	*0*	*1*	*0*	*1*	*0*	*1*	*0.25*
0	*1*	*0*	*0*	*1*	*1*	*0*	*0.25*
0	*1*	*1*	*0.25*	*1*	*1*	*1*	*0*

For this example, we observe a strong conditional dependence. One easily verifies, for example, that $\mathbf{p}^{(1)}_{x_1=1}(0)\mathbf{p}^{(1)}_{x_1=1}(0) = 0.5 \times 0.5 = 0.25$, while the joint probability is $\mathbf{p}_{x_1=1}(0,0) = 0.5$. One can even speak of a kind of deterministic dependence, since $y_1 = y_2$ for $x_1 = 0$ and $y_2 = 1 - y_1$ for $x_1 = 1$. However, the labels are unconditionally independent. In fact, noting that the marginals are given by $\mathbf{p}^{(1)}(1) = \mathbf{p}^{(2)}(1) = 0.5$, the joint probability is indeed the product of marginals.

The next two sections discuss unconditional and conditional dependence more in detail. We will claim that exploiting both types of dependence can improve the generalization performance of a multi-label classifier.

4. Unconditional Label Dependence

Researchers speaking about dependencies or correlations between labels are typically referring to unconditional dependence. Empirically, this type of dependence can be measured in terms of correlation coefficients, such as Pearson correlation, or any other type of statistical measure of (in)dependence on the observed labels in the training data.

4.1. Modeling Unconditional Dependence

For a better understanding of unconditional dependence between labels, it is convenient to connect multi-label classification with multivariate regression (often called multi-output regression), i.e., the simultaneous prediction of several real-valued variables. Historically, multi-label classification has indeed been treated as a specific case of multivariate regression in statistics, for example within the context of vector generalized linear models (VGLMs), as summarized in (Song, 2007, chapter 6) and (Izenman, 2008, chapter 6).

In a regression context, unconditional dependence between two continuous labels Y_i and Y_j means that $\mathbb{E}[Y_i \mid Y_j] \neq \mathbb{E}[Y_i]$. Let us adopt the standard statistical notation for describing a multivariate regression model, namely

$$Y_i = h_i(\boldsymbol{X}) + \epsilon_i(\boldsymbol{X}) \tag{10}$$

for all $i = 1, ..., m$, where the functions $h_i : \mathcal{X} \to \mathbb{R}$ represent a set of m parameterized models and the random variables $\epsilon_i(\boldsymbol{x})$ a set of m error terms satisfying

$$\mathbb{E}[\epsilon_i(\boldsymbol{x})] = 0 \tag{11}$$

for all $\boldsymbol{x} \in \mathcal{X}$ and $i = 1, ..., m$. Remark that MLC can be considered as a specific case of the multivariate regression model (10), where the labels Y_i are binary instead of continuous random variables; then, however, the assumption (11) is typically violated.

In general, the distribution of the noise terms can depend on $\boldsymbol{x}$. Moreover, two noise terms ϵ_i and ϵ_j can

also depend on each other. However, unconditional dependence between labels more often originates from dependencies between the underlying models $h_i(\cdot)$, i.e., the deterministic parts of the model (10). Roughly speaking, if there is a function $f(\cdot)$ such that $h_i \approx f \circ h_j$ in the sense that

$$h_i(\boldsymbol{x}) = f(h_j(\boldsymbol{x})) + g(\boldsymbol{x}) \ , \tag{12}$$

with $g(\cdot)$ being "negligible" in the sense that $g(\boldsymbol{x}) \approx 0$ with high probability (or, say, for most $\boldsymbol{x}$), then this "f-dependence" between h_i and h_j is likely to dominate the averaging process in (9), whereas $g(\cdot)$ and the error terms ϵ_i will play a less important role (or simply cancel out). In other words, the dependence between h_i and h_j, despite being only probable and approximate, will induce a dependence between the labels Y_i and Y_j.

This global dependence is a constraint that can be used by a learning algorithm for the purpose of regularization. This way, it may indeed help to improve predictive accuracy.

4.2. Example

Consider a simple problem with a two-dimensional input $\boldsymbol{x} = (x_1, x_2)$ uniformly distributed in $[-1, +1] \times [-1, +1]$, and two labels Y_1, Y_2 distributed as follows: $Y_1 = [\![x_1 > 0]\!]$, i.e., the first label is just the sign of the first input attribute. The second label is defined in the same way, but the decision boundary ($x_1 = 0$) is rotated by an angle $\alpha \in [0, \pi]$. The two decision boundaries partition the input space into four regions C_{ij} identified by $i = Y_1$ and $j = Y_2$. Moreover, the two error terms shall be independent and both flip the label with a probability 0.1 (i.e., $\epsilon_1 = 0$ with probability 0.9 and $\epsilon_1 = 1 - 2[\![x_1 > 0]\!]$ with probability 0.1); see Fig. 1 for a typical data set.

For α close to 0, the two labels are almost identical, whereas for $\alpha = \pi$, they are orthogonal to each other. More specifically, (12) holds with $f(\cdot)$ the identity and $g(\boldsymbol{x})$ given by ± 1 in the "overlap regions" C_{01} and C_{10} (shaded in gray) and 0 otherwise.

We compare two learning methods on this problem. The first one is the simple nearest neighbor (1NN) classifier. The second one is a kind of stacking variant (cf. Section 4.3 below), in which unconditional dependence is exploited by training another classifier on top of 1NN estimation. The idea is to look for "correction rules" of the following form: If the original prediction is (y_1, y_2), then the true labeling is (y_1^{adj}, y_2^{adj}). In our case, given the small number of labels, a classifier of this kind can simply be expressed in tabular form. More specifically, we train two classifiers

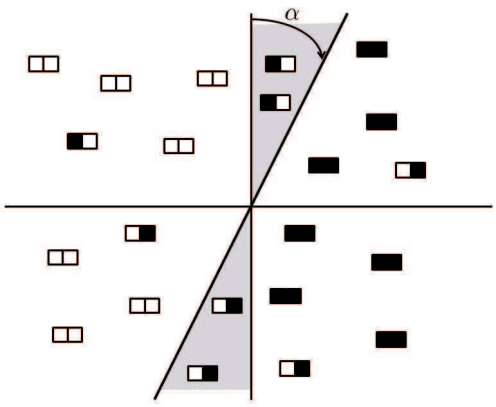

Figure 1. Exemplary data set: The two labels are encoded as neighbored squares, colored in black for positive and white for negative.

$h_1^{adj}, h_2^{adj} : \{0,1\}^2 \to \{0,1\}$, where h_1 (h_2) outputs the most probable value of the first (second) label, given a labeling (y_1, y_2) as an original prediction; probability is estimated by relative frequency.

Fig. 2 shows the performance curves of the two methods as a function of α, where performance corresponds to the expected error rate given a random training set of size 50; two types of error are considered, the subset 0/1 loss (3) and the Hamming loss (4). As can be seen from these curves, unconditional dependence can indeed help to improve performance, at least as long as $Y_1 \approx Y_2$, i.e., for small α. If the (functional) dependence between $h_1(\cdot)$ and $h_2(\cdot)$ becomes weaker, however, this advantages disappears.

One should of course realize that unconditional dependence is global information that holds on average but not necessarily in a single point $\boldsymbol{x} \in \mathcal{X}$. Thus, despite being beneficial on average, imposing corresponding properties on single instances or biasing the prediction toward the average may prevent from inferring locally (Bayes) optimal solutions.

4.3. Learning Algorithms

There are many learning algorithms that are able to exploit unconditional label dependence, especially when treating multi-label classification as a specific case of multivariate regression. In the section, we give a brief overview of such algorithms, though without claiming completeness. Roughly speaking, most algorithms seek to reduce the variance of the predictions obtained by models that ignore unconditional label dependence and instead train one model for each label independently of all other labels, such as binary rele-

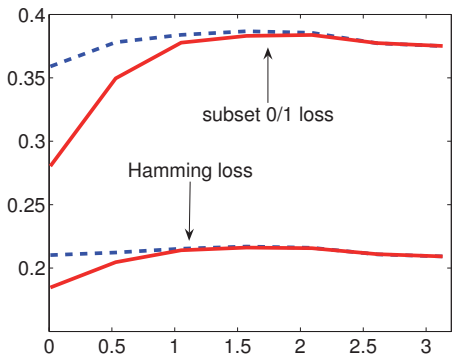

Figure 2. Learning curves for 1NN estimation (dashed line) and stacking (solid): error rate as a function of the angle α defining the decision boundary of the second label.

vance in MLC or separate least-squares estimators in multivariate regression.

Stacking. Methods like stacking (Cheng and Hüllermeier, 2009) or the C&W procedure (Breiman and Friedman, 1997) replace the original predictions, obtained by learning every label separately, by correcting them in light of information about the predictions of the other labels. For example, if the initial prediction for the first label of an instance $\boldsymbol{x}$ is given by $h_1^{\text{ini}}(\boldsymbol{x})$, then the adjusted prediction $h_1^{\text{adj}}(\boldsymbol{x})$ may take the following form:

$$h_1^{\text{adj}}(\boldsymbol{x}) = a_1 h_1^{\text{ini}}(\boldsymbol{x}) + \sum_{k=2}^{m} a_k h_k^{\text{ini}}(\boldsymbol{x})$$

This transformation of the initial prediction should be interpreted as a regularization procedure: a bias is introduced, in an attempt to decrease the variance. Similar effects can of course also be achieved by other means, for example by using prior information in Bayesian inference (Zhang and Zhou, 2007).

Reduced-rank regression. Instead of adjusting the predictions in a post-processing step, one can of course think of including a label-based regularization in the algorithm itself. This is the motivation behind reduced-rank regression (RRR) and related methods (Izenman, 1975). RRR has been introduced in statistics more than thirty years ago for multivariate regression tasks where outputs are unconditionally dependent. Under the assumption that every label can be represented as a linear model of the inputs, i.e.,

$$h_k(\boldsymbol{x}) = \mathbf{w}_k \cdot \boldsymbol{x}\,,$$

with $\mathbf{w}_k$ a vector of parameters that is specific for every label, the RRR method introduces a regularization term in the least-squares objective function, penalizing for the rank of the matrix $\mathbf{W} = (\mathbf{w}_1, ..., \mathbf{w}_m)$. Given the restriction that the rank of $\mathbf{W}$ is $r < m$, the predictive model is forced toward predictions of label vectors having similar or even identical components, resulting in a similar effect as stacking.

Multi-task learning. Explicit modeling of label dependence has also played a key role in the development of related areas like multi-task learning and transfer learning, where task i with model h_i and task j with model h_j are assumed to be related (Caruana, 1997). Similar types of label-based regularization characterize recent developments in these domains. Amongst others, the regularized multi-task learning algorithm of (Evgeniou and Pontil, 2004) considers parameter vectors of the type

$$\mathbf{w}_k = \mathbf{w}_0 + \mathbf{v}_k\,,$$

with the assumption that, for related tasks, the norms of the label-specific vectors $\mathbf{v}_k$ are small compared to the norm of the base parameter vector $\mathbf{w}_0$.

Label dimensionality reduction. Methods like kernel dependency estimation (Weston et al., 2002) and its variants (Yu et al., 2006; Hsu et al., 2009) originate from a different point of view, but still perform a very similar job for predicting multivariate responses. Kernel dependency estimation consists of a three-step procedure. The first step conducts a principal component analysis of the label space to make the labels uncorrelated. As an alternative, one can also opt to use kernel PCA instead of regular PCA in this step, for deriving non-linear combinations of the labels or for predicting structured outputs. Subsequently, the transformed labels (i.e. the principal components) are used in a simple multivariate regression method that does not have to care about label dependencies, knowing that the transformed labels are uncorrelated. In the last step, the predicted labels of test data are transformed again to the original label space. Label-based regularization can be included in this approach as well, simply by using only the first $r < m$ principal components in steps two and three, similar to regularization based on feature selection in methods like principal component regression (Hastie et al., 2009).

5. Conditional Label Dependence

As mentioned previously, unconditional dependence is a kind of expected dependence, averaging over the marginal distribution of the input $\boldsymbol{X}$. As opposed to this, conditional dependence refers to the dependence of the labels given a fixed instance $\boldsymbol{x}$ in the feature

space. Roughly speaking, while unconditional dependence mostly concerns the deterministic part in (10), i.e., the functions $h_i(\cdot)$, conditional dependence concerns the stochastic part, i.e., the error terms ϵ_i. Indeed, once an instance $\boldsymbol{x}$ has been observed, the $h_i(\boldsymbol{x})$ are simply constants and thus become irrelevant for any kind of dependency analysis.

5.1. Modeling Conditional Dependence

The posterior probability distribution $\mathbf{p}_{\boldsymbol{x}}(\mathbf{Y})$ provides a convenient point of departure for analyzing conditional label dependence, since it informs about the probability of each label combination as well as the marginal probabilities. In a stochastic sense, there is a dependency between the labels if the joint conditional distribution is not the product of the marginals (like in the above example).

For instance, in our example from Section 4.2, conditional independence between Y_1 and Y_2 follows from the assumption of independent error terms ϵ_1 and ϵ_2. This independence is lost, however, when assuming a close dependency between the error terms, namely $\epsilon_1 = \epsilon_2$. In fact, even though the marginals will remain the same, the joint distribution will change in that case. The following table compares the two distributions for an instance $\boldsymbol{x}$ form the region C_{11}:

$\mathbf{p}_{\boldsymbol{x}}(\mathbf{Y})$	0	1	$\mathbf{p}^{(1)}_{\boldsymbol{x}}(Y_1)$
0	0.01 | 0.10	0.09 | 0.00	0.10
1	0.09 | 0.00	0.81 | 0.90	0.90
$\mathbf{p}^{(1)}_{\boldsymbol{x}}(Y_2)$	0.10	0.90	1

A connection with multivariate regression can be made by defining error terms in (10) in a proper way. In terms of their expectation, we have

$$\mathbb{E}[\epsilon_i(\boldsymbol{x})] = \begin{cases} \mathbf{p}^{(i)}_{\boldsymbol{x}}(1) & \text{if } h_i(\boldsymbol{x}) = 0\,, \\ -\mathbf{p}^{(i)}_{\boldsymbol{x}}(0) & \text{if } h_i(\boldsymbol{x}) = 1\,, \end{cases}$$

for $i = 1, ..., m$ and

$$\mathbb{E}[\epsilon_i(\boldsymbol{x})\epsilon_j(\boldsymbol{x})] = \begin{cases} \mathbf{p}^{(i,j)}_{\boldsymbol{x}}(1,1) \text{ if } h_i(\boldsymbol{x})=0,\ h_j(\boldsymbol{x})=0, \\ -\mathbf{p}^{(i,j)}_{\boldsymbol{x}}(1,0) \text{ if } h_i(\boldsymbol{x})=0,\ h_j(\boldsymbol{x})=1, \\ -\mathbf{p}^{(i,j)}_{\boldsymbol{x}}(0,1) \text{ if } h_i(\boldsymbol{x})=1,\ h_j(\boldsymbol{x})=0, \\ \mathbf{p}^{(i,j)}_{\boldsymbol{x}}(0,0) \text{ if } h_i(\boldsymbol{x})=1,\ h_j(\boldsymbol{x})=1, \end{cases}$$

for $i, j = 1, ..., m$. This observation implies the following proposition that directly links multi-label classification and multivariate regression: Two labels Y_i and Y_j are conditionally dependent given $\boldsymbol{x}$ if and only if the error terms $\epsilon_i(\boldsymbol{x})$ and $\epsilon_j(\boldsymbol{x})$ in (10) are conditionally dependent, i.e., $\mathbb{E}[\epsilon_i(\boldsymbol{x})\epsilon_j(\boldsymbol{x})] \neq \mathbb{E}[\epsilon_i(\boldsymbol{x})]\mathbb{E}[\epsilon_j(\boldsymbol{x})]$.

5.2. Loss Functions and Risk Minimization

Can conditional label dependence help to improve the generalization performance in MLC? Unlike the case of unconditional label dependence, the answer is more nuanced in this case and has not been widely studied so far. The results presented in (Dembczyński et al., 2010) suggest that, despite being affirmative in general, the answer strongly depends on the loss function to be minimized.

Recall that the risk-minimizing model $\mathbf{h}^*$ is given by

$$\mathbf{h}^*(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}} \mathbb{E}_{\mathbf{Y}|\mathbf{X}} L(\mathbf{Y}, \boldsymbol{y}), \tag{13}$$

where $L(\mathbf{Y}, \boldsymbol{y})$ is a loss function defined on multi-label predictions. Now, let us take a look at three commonly used loss functions in multi-label problems, namely the Hamming loss, rank loss, and the subset 0/1 loss as defined in Section 2.

- For the subset 0/1 loss (3), it is easy to see that the risk-minimizing prediction is given by the mode of the distribution:

$$\mathbf{h}^*(\boldsymbol{x}) = \arg\max_{\boldsymbol{y}\in\mathcal{Y}} \mathbf{p}_{\boldsymbol{x}}(\boldsymbol{y}) \ . \tag{14}$$

- For the Hamming loss (4), it is easy to see that (13) is obtained by

$$h^*_i(\boldsymbol{x}) = \arg\max_{b\in\{0,1\}} \mathbf{p}^{(i)}_{\boldsymbol{x}}(b). \tag{15}$$

- As shown by (Dembczyński et al., 2010), the expected rank loss (5) can be minimized by sorting the labels according to their probability of relevance. In other words, following ranking function is a risk minimizer for (5):

$$f_i(\boldsymbol{x}) \ = \ \mathbf{p}^{(i)}_{\boldsymbol{x}}(1) \tag{16}$$

As an important consequences of the above results we note that, according to (15) and (16), a risk-minimizing prediction for the Hamming and the rank loss can be obtained from the marginal distributions $\mathbf{p}^{(i)}_{\boldsymbol{x}}(Y_i)$ $(i = 1, \ldots, m)$ alone. In other words, it is not necessary to know the joint label distribution $\mathbf{p}_{\boldsymbol{x}}(\mathbf{Y})$ on $\mathcal{Y}$. Or, stated differently, risk-minimizing predictions can in principle be made without any knowledge about the dependency between labels.

As opposed to this, (14) shows that the mode of the entire distribution of $\mathbf{Y}$ given $\mathbf{X}$ is needed to minimize the subset zero-one loss. In other words, the derivation of a risk-minimizing prediction requires the modeling of the joint distribution, and hence the modeling of dependencies between labels.[2]

[2] Let us remark, however, that the joint mode coincides

5.3. Learning Algorithms

The previous findings suggest that modeling conditional dependence is reasonable when the joint conditional distribution is needed, or any property of this distribution that cannot be easily derived from its marginals. In this section, we discuss some algorithms that are able to deal with this problem. Since inferring the entire joint distribution or any function thereof (like the mode or an optimal response with respect to a given loss function) can become costly, we shall also comment on complexity issues.

Label powerset classifier. The label powerset classifier (LPC) reduces MLC to multi-class classification. To this end, it considers each label subset $L \in \mathcal{L}$ as a distinct meta-class (Tsoumakas and Katakis, 2007; Tsoumakas and Vlahavas, 2007). The number of these meta-classes may become as large as $|\mathcal{L}| = 2^m$, which is an obvious disadvantage of this approach.

Since prediction of the most probable meta-class is equivalent to prediction of the mode of the joint label distribution, LPC is tailored for the subset 0/1 loss. Theoretically, it can even deliver an estimate of the joint distribution provided the underlying multi-class classifier is in fact a probability estimator. Practically, however, the large number of meta-classes makes probability estimation an extremely difficult problem. In this regard, we also mention that most implementations of LPC essentially ignore label combinations that are not presented in the training set or, stated differently, tend to underestimate (set to 0) their probabilities.

Probabilistic classifier chains. In order to avoid the high complexity of LPC, one can exploit the product rule of probability (8). More specifically, to estimate the joint distribution of labels, one possibility is to learn m functions $g_i(\cdot)$ on augmented input spaces $\mathcal{X} \times \{0,1\}^{i-1}$, respectively, taking $y_1, \ldots, y_{i-1}$ as additional attributes:

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

Here, we assume that the function $g_i(\cdot)$ can be interpreted as a probabilistic classifier whose prediction is the probability that $y_i = 1$, or at least a reasonable approximation thereof. This approach is referred to as probabilistic classifier chains (Dembczyński et al., 2010). As it essentially comes down to training m classifiers (in augmented feature spaces), its computational complexity is acceptable.

Much more problematic, however, is inference from the given joint distribution. In fact, since exact inference can become infeasible, approximate methods may have to be used. For example, a simple greedy approximation of the mode is obtained by successively choosing the most probable label according to each of the classifiers' predictions. This approach has been introduced in (Read et al., 2009), albeit without a probabilistic interpretation.

Conditional random fields. Conditional dependence can be represented in terms of graphical models. For example, the conditionally trained undirected graphical models, or conditional random fields (CRF) for short, have been used for dealing with multi-label problems in (Ghamrawi and McCallum, 2005). Such models provide a rich framework for representing relationships between labels and features of a given domain.

If the nature of the conditional dependence is known in advance, the use of graphical structures for modeling and learning seems to be the most adequate solution. The output of such a model is an estimate of the entire joint distribution of labels. The learning can be performed in terms of gradient descent methods. Of course, the cost of learning depends on the complexity of the modeled structure. Usually, the interactions of a low degree (like pairwise) are enough to obtain a good solution.

In methods such as PCC and CRF, we meet the problem of inference from the estimated joint distribution. This limits the applicability of these methods to data sets with a small to moderate number of labels, say, not more than about 15. There are, however, possibilities to develop approximate inference schemes that trade off accuracy against efficiency in a reasonable way (Ghamrawi and McCallum, 2005). This can be done in different ways, for example by limiting the inference only to the label combinations that appear in the training set (as usually done by LPC), 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).

AdaBoost.LC. Instead of estimating the joint probability distribution, one can also try to minimizes a given loss function in a more direct way. In (Amit et al., 2007), so-called label covering loss functions are introduced, including Hamming and the

with the marginal modes under specific conditions, for example in the case of label m-independence or if the probability of the joint mode is greater or equal 0.5.

subset 0/1 losses as special cases. The authors also propose a learning algorithm suitable for minimizing covering losses, called AdaBoost.LC. This algorithm is based on boosting and yields a single vector $\boldsymbol{h}(\boldsymbol{x})$ as a prediction.

Kernelized loss functions. The last concept we mention is kernelized loss functions. As remarked in (Weston et al., 2002), a loss function can be seen as a measure of similarity in the output space. Since a kernel function is also a similarity function, one can easily generalize loss functions to the concept of kernels. Consequently, kernel dependence estimation (Weston et al., 2002) as well as structural SVM (Tsochantaridis et al., 2005) are also able to deal with conditional dependence.

6. Conclusions

The goal of this paper is to clarify some issues related to label dependence in multi-label classification. Even though most of the 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 correlations, should be considered with diligence. First, different types of dependence can be considered; in this paper, we focused on the distinction between conditional and unconditional dependence. Second, the question whether or not label dependencies can be harnessed to improve predictive accuracy cannot be answered in a blanket way. Instead, it depends on the concrete goal of the learning problem (as expressed, for example, in terms of a loss function to be minimized),

References

Y. Amit, O. Dekel, and Y. Singer. A boosting algorithm for label covering in multilabel problems. In *JMLR W&P*, volume 2, pages 27–34, 2007.

L. Breiman and J. Friedman. Predicting multivariate responses in multiple linear regression. *J. of the Royal Statistical Society: Series B*, 69:3–54, 1997.

R. Caruana. Multitask learning: A knowledge-based source of inductive bias. *Machine Learning*, 28:41–75, 1997.

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

K. Dembczyński, W. Cheng, and E. Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proc. ICML 2010*, Haifa, Israel, 2010. to appear.

T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117, 2004.

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

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, Second Edition*. Springer, 2009.

D. Hsu, S. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *Advances in Neural Information Processing Systems*, 2009.

A. Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5:248–262, 1975.

A. Izenman. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer, 2008.

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

P. Song. *Correlated Data Analysis: Modeling Analytics and applications*. Springer, 2007.

Y. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and independent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

G. Tsoumakas and I. Katakis. Multi label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

G. Tsoumakas and I. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *Proceedings of the 18th European Conference on Machine Learning (ECML 2007)*, pages 406–417, Warsaw, Poland, September 2007.

J. Weston, O. Chapelle, A. Elisseeff, B. Schoelkopf, and V. Vapnik. Kernel dependency estimation. In *Advances in Neural Information Processing Systems*, 2002.

S. Yu, K. Yu, V. Tresp, and H.-P. Kriegel. Multi-output regularized feature projection. *IEEE Trans. on Knowl. and Data Eng.*, 18(12):1600–1613, 2006.

M.-L. Zhang and Z.-H. Zhou. ML-kNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.