# Learning Interpretable Rules for Multi-label Classification

Preprint version. To be published in: Explainable and Interpretable Models in Computer Vision and Machine Learning. The Springer Series on Challenges in Machine Learning. Springer (2018)

Eneldo Loza Mencía , Johannes Fürnkranz, Eyke Hüllermeier, Michael Rapp

**Abstract** Multi-label classification (MLC) is a supervised learning problem in which, contrary to standard multiclass classification, an instance can be associated with several class labels simultaneously. In this chapter, we advocate a rule-based approach to multi-label classification. Rule learning algorithms are often employed when one is not only interested in accurate predictions, but also requires an interpretable theory that can be understood, analyzed, and qualitatively evaluated by domain experts. Ideally, by revealing patterns and regularities contained in the data, a rule-based theory yields new insights in the application domain. Recently, several authors have started to investigate how rule-based models can be used for modeling multi-label data. Discussing this task in detail, we highlight some of the problems that make rule learning considerably more challenging for MLC than for conventional classification. While mainly focusing on our own previous work, we also provide a short overview of related work in this area.

**Key words:** Multi-label classification, label-dependencies, rule learning, separate-and-conquer

Eneldo Loza Mencía
Technische Universität Darmstadt, Knowledge Engineering Group,
e-mail: eneldo@ke.tu-darmstadt.de

Johannes Fürnkranz
Technische Universität Darmstadt, Knowledge Engineering Group,
e-mail: juffi@ke.tu-darmstadt.de

Eyke Hüllermeier
Universität Paderborn, Intelligent Systems, e-mail: eyke@upb.de

Michael Rapp
Technische Universität Darmstadt, Knowledge Engineering Group,
e-mail: mrapp@ke.tu-darmstadt.de

## 1 Introduction

Multi-label classification (MLC) is a problem in the realm of supervised learning. Contrary to conventional, single-label classification, MLC allows an instance to be associated with multiple class labels simultaneously. Dealing with and taking advantage of (statistical) dependencies between the presence and absence (relevance and irrelevance) of different labels has been identified as a key issue in previous work on MLC. To improve predictive performance, essentially all state-of-the-art MLC algorithms therefore seek to capture label dependencies in one way or the other.

In this chapter, we will argue that inductive rule learning is a promising approach for tackling MLC problems. In particular, rules provide an interpretable model for mapping inputs to outputs, and allow for tightly integrating input variables and labels into coherent comprehensible theories. For example, so-called global dependencies between labels can be explicitly modeled and expressed in the form of rules. Moreover, these can be easily generalized to local dependencies, which include regular input features as a local context in which a label dependency holds. Such rules, which mix labels and features, are nevertheless directly interpretable and comprehensible for humans. Even if complex and long rules are generated, the implication between labels can be easily grasped by focusing on the part of the rules that actually considers the labels. Hence, in contrast to many other model types that capture label dependencies implicitly, such dependencies can be analyzed and interpreted more directly.

We will start with a brief definition and formalization of the multi-label learning problem (Section 2), in which we also introduce a dataset that will serve as a running example. In Section 3, we then define multi-label rules, highlighting the differences to conventional classification rules, discuss various dimensions and choices that have to be made, and list some challenges for learning such rules. Sections 4 and 5 then deal with descriptive and predictive multi-label rule learning, respectively. The former recalls association-rule based approaches and discusses how properties like anti-monotonicity can be used to efficiently search for a suitable head for a given rule body, whereas the latter discusses two approaches for learning predictive rule-based theories: one based on stacking different label prediction layers, and another one based on adapting the separate-and-conquer or covering strategy to the multi-label case. Finally, in Section 6, we present and discuss rule-based theories for a few well-known sample multi-label databases, before we conclude in Section 7.

## 2 Multi-Label Classification

Multi-label classification has received a lot of attention in the recent machine learning literature (Tsoumakas et al., 2010, 2012; Gibaja and Ventura, 2014, 2015; Varma and Cissé, 2015; Herrera et al., 2016; Zhang and Zhou, 2014). The motivation for MLC originated in the field of text categorization (Hayes and Weinstein, 1991; Lewis, 1992, 2004), but nowadays multi-label methods are used in applications as

diverse as music categorization (Trohidis et al., 2008), semantic scene classification (Boutell et al., 2004), or protein function classification (Elisseeff and Weston, 2001).

## *2.1 Problem Definition*

The task of MLC is to associate an instance with one or several labels $\lambda_i$ out of a finite label space $\mathscr{L}$. with $n = |\mathscr{L}|$ being the number of available labels. Contrary to ordinary classification, MLC allows each instance to be associated with more than one (class) label, but, in contrast to multiclass learning, alternatives are not assumed to be mutually exclusive, such that multiple labels may be associated with a single instance. Figure 1(a) shows an example, which relates persons described with some demographic characteristics to the newspapers and magazines they subscribe. Obviously, the number of subscriptions can vary. For example, subject #1 (a single male with primary education and no kids) has subscribed to no magazines at all, whereas #13 (a divorced male with university degree and children) obtains a quality newspaper and a tabloid.

Potentially, there are $2^n$ different allowed allocations of the output space, which is a dramatic growth compared to the $n$ possible states in the multiclass setting. However, not all possible combinations need to occur in the database. For example, nobody in this database has subscribed to both a fashion and a sports magazine. Note that these label attributes are not independent. The fact that there may be correlations and dependencies between the labels in $\mathscr{L}$ makes the multi-label setting particularly challenging and interesting compared to the classical setting of binary and multiclass classification.

Formally, MLC refers to the task of learning a predictor $f : \mathscr{X} \rightarrow 2^{\mathscr{L}}$ that maps elements $\mathbf{x}$ of an instance space $\mathscr{X}$ to subsets $P$ of a set of labels $\mathscr{L} = \{\lambda_1, \ldots, \lambda_n\}$. Equivalently, predictions can be expressed as binary vectors $\mathbf{y} = f(\mathbf{x}) = (y_1, \ldots, y_n) \in \{0,1\}^n$, where each attribute $y_i$ encodes the presence (1) or absence (0) of the corresponding label $\lambda_i$, We will use these two notations interchangeably, i.e., $y$ will be used to refer to an element in a binary prediction vector, whereas $\lambda$ refers to an element in a predicted label set.

An instance $\mathbf{x}_j$ is in turn represented in attribute-value form, i.e., it consists of a vector $\mathbf{x}_j \coloneqq \langle x_1, ..., x_a \rangle \in \mathscr{X} = \phi_1 \times ... \times \phi_a$, where $\phi_i$ is a numeric or nominal attribute.

Consequently, the training data set of an MLC problem can be defined as a sequence of tuples $\mathscr{T} \coloneqq \langle (\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_m, \mathbf{y}_m) \rangle \subseteq \mathscr{X} \times \mathscr{Y}$ with $m = |\mathscr{T}|$. Figure 1 shows both representations, once with sets as outputs (1(a)) and once with binary vectors as outputs (1(b)).

| Person | | | | | Subscribed Magazines |
|---|---|---|---|---|---|
| **No.** | **Education** | **Marital** | **Sex** | **Children?** | |
| 1 | Primary | Single | Male | No | $\emptyset$ |
| 2 | Primary | Single | Male | Yes | $\emptyset$ |
| 3 | Primary | Married | Male | No | {tabloid} |
| 4 | University | Divorced | Female | No | {quality, fashion} |
| 5 | University | Married | Female | Yes | {quality, fashion} |
| 6 | Secondary | Single | Male | No | {tabloid} |
| 7 | University | Single | Male | No | {quality, tabloid} |
| 8 | Secondary | Divorced | Female | No | {quality, sports} |
| 9 | Secondary | Single | Female | Yes | {tabloid, fashion} |
| 10 | Secondary | Married | Male | Yes | {quality, tabloid} |
| 11 | Primary | Married | Female | No | $\emptyset$ |
| 12 | Secondary | Divorced | Male | Yes | $\emptyset$ |
| 13 | University | Divorced | Male | Yes | {quality, tabloid} |
| 14 | Secondary | Divorced | Male | No | {quality, sports} |

(a) With set-valued outputs

| Person | | | | | Subscribed Magazines | | | |
|---|---|---|---|---|---|---|---|---|
| **No.** | **Education** | **Marital** | **Sex** | **Children?** | **Quality** | **Tabloid** | **Fashion** | **Sports** |
| 1 | Primary | Single | Male | No | 0 | 0 | 0 | 0 |
| 2 | Primary | Single | Male | Yes | 0 | 0 | 0 | 0 |
| 3 | Primary | Married | Male | No | 0 | 1 | 0 | 0 |
| 4 | University | Divorced | Female | No | 1 | 0 | 1 | 0 |
| 5 | University | Married | Female | Yes | 1 | 0 | 1 | 0 |
| 6 | Secondary | Single | Male | No | 0 | 1 | 0 | 0 |
| 7 | University | Single | Male | No | 1 | 1 | 0 | 0 |
| 8 | Secondary | Divorced | Female | No | 1 | 0 | 0 | 1 |
| 9 | Secondary | Single | Female | Yes | 0 | 1 | 1 | 0 |
| 10 | Secondary | Married | Male | Yes | 1 | 1 | 0 | 0 |
| 11 | Primary | Married | Female | No | 0 | 0 | 0 | 0 |
| 12 | Secondary | Divorced | Male | Yes | 0 | 0 | 0 | 0 |
| 13 | University | Divorced | Male | Yes | 1 | 1 | 0 | 0 |
| 14 | Secondary | Divorced | Male | No | 1 | 0 | 0 | 1 |

(b) With binary output vectors

**Fig. 1** Two representations of a sample multi-label classification problem, which relates demographic characteristics to subscribed newspapers and magazines.

## *2.2 Dependencies in Multi-Label Classification*

The simplest and best known approach to multi-label classification is *binary relevance* (BR) learning (e.g. Tsoumakas et al., 2010). It tackles a multi-label problem by learning one classifier for each label, using all its instances as positive and the others as negative examples. The obvious disadvantage of this transformation is the ignorance of possible dependencies between the labels. More advanced methods seek to exploit such dependencies, mainly with the goal of improving predictive accuracy.

The goal of most classification algorithms is to capture dependencies between input variables $x_j$ and the output variables $y_i$. In fact, the prediction $\hat{y} = f(\mathbf{x})$ of a scoring classifier $f$ is often regarded as an approximation of the conditional probability $\Pr(y = \hat{y} \mid \mathbf{x})$, i.e., the probability that $\hat{y}$ is the true label for the given instance $\mathbf{x}$. In MLC, dependencies may not only exist between $\mathbf{x}$ and each target $y_i$, but also between the labels $y_1, \ldots, y_n$ themselves.

A key distinction is between *unconditional* and *conditional independence* of labels. In the first case, the joint distribution $\Pr(\mathbf{y})$ in the label space factorizes into the product of the marginals $\Pr(y_i)$, i.e., $\Pr(\mathbf{y}) = \Pr(y_1) \cdots \Pr(y_n)$, whereas in the latter case, the factorization $\Pr(\mathbf{y} \mid \mathbf{x}) = \Pr(y_1 \mid \mathbf{x}) \cdots \Pr(y_n \mid \mathbf{x})$ holds conditioned on $\mathbf{x}$, for every instance $\mathbf{x}$. In other words, unconditional dependence is a kind of global dependence (for example originating from a hierarchical structure on the labels), whereas conditional dependence is a dependence locally restricted to a single point in the instance space.

In the literature, both types of dependence have been explored. For example, Sucar et al. (2014) model label dependence in the form of a Bayesian network. Chekina et al. (2013) provide an empirical analysis of the different types of dependencies between pairs of labels on standard benchmark datasets, and analyze the usefulness of modeling them. Unconditional dependencies were analyzed by a simple $\chi^2$ test on the label co-occurrence matrix, whereas for detecting unconditional dependencies they compared the performance of a classifier $f_i$ for a label $y_i$ trained on the instance features $(\mathbf{x})$ to the same learning algorithm being applied to the input space $(\mathbf{x}, y_j)$ augmented by the label feature of a second label $y_j$. If the predictions differ statistically significantly, then $y_i$ is assumed to be conditionally dependent on $y_j$. Their evaluations show that pairwise unconditional dependencies occur more frequently than pairwise conditional dependencies, and that, surprisingly, modeling global dependencies is more beneficial in terms of predictive performance. However, this finding is very specific to their setting, where the dependence information is basically used to guide a decomposition into smaller problems with less labels that are either independent or dependent. In addition, only pairwise co-occurrence and pairwise exclusion can effectively be exploited by their approach. As we will see in Section 3.2, rules can be used to flexibly formulate a variety of different dependencies, including partially label-dependent or local dependencies.

## *2.3 Evaluation of Multi-label Predictions*

### 2.3.1 Bipartition Evaluation Functions

To evaluate the quality of multi-label predictions, we use bipartition evaluation measures (cf. Tsoumakas et al. (2010)) which are based on evaluating differences between true (*ground truth*) and predicted label vectors. They can be considered as functions of two-dimensional *label confusion matrices* which represent the *true positive* ($TP$), *false positive* ($FP$), *true negative* ($TN$) and *false negative* ($FN$) label pre-

dictions. For a given example $\mathbf{x}_j$ and a label $y_i$ the elements of an atomic confusion matrix $C_i^j$ are computed as

$$C_i^j = \begin{pmatrix} TP_i^j & FP_i^j \\ FN_i^j & TN_i^j \end{pmatrix} = \begin{pmatrix} y_i^j \hat{y}_i^j & (1-y_i^j)\hat{y}_i^j \\ (1-y_i^j)(1-\hat{y}_i^j) & y_i^j(1-\hat{y}_i^j) \end{pmatrix} \tag{1}$$

where the variables $y_i^j$ and $\hat{y}_i^j$ denote the absence (0) or presence (1) of label $\lambda_i$ of example $\mathbf{x}_j$ according to the ground truth or the predicted label vector, respectively.

Note that for candidate rule selection we assess $TP$, $FP$, $TN$, and $FN$ differently. To ensure that absent and present labels have the same impact on the performance of a rule, we always count correctly predicted labels as $TP$ and incorrect predictions as $FP$, respectively. Labels for which no prediction is made are counted as $TN$ if they are absent, or as $FN$ if they are present.

### 2.3.2 Multi-Label Evaluation Functions

In the following some of the most common bipartition metrics $\delta(C)$ used for MLC are presented (cf., e.g., Tsoumakas et al. (2010)). They are mappings $\mathbb{N}^{2x2} \rightarrow \mathbb{R}$ that assign a real-valued score (often normalized to $[0,1]$) to a confusion matrix $C$. Predictions that reach a greater score outperform those with smaller values.

- **Precision:** Percentage of correct predictions among all predicted labels.

$$\delta_{prec}(C) \coloneqq \frac{TP}{TP+FP} \tag{2}$$

- **Hamming accuracy:** Percentage of correctly predicted present and absent labels among all labels.

$$\delta_{hamm}(C) \coloneqq \frac{TP+TN}{TP+FP+TN+FN} \tag{3}$$

- **F-measure:** Weighted harmonic mean of precision and recall. If $\beta < 1$, precision has a greater impact. If $\beta > 1$, the F-measure becomes more recall-oriented.

$$\delta_F(C) \coloneqq \frac{\beta^2+1}{\frac{\beta^2}{\delta_{rec}(C)} + \frac{1}{\delta_{prec}(C)}} \text{ , with } \delta_{rec}(C) = \frac{TP}{TP+FN} \text{ and } \beta \in [0,\infty] \tag{4}$$

- **Subset accuracy:** Percentage of perfectly predicted label vectors among all examples. Per definition, it is always calculated using example-based averaging.

$$\delta_{acc}(C) \coloneqq \frac{1}{m}\sum_j [\mathbf{y}_j = \hat{\mathbf{y}}_j] \text{ , with } [x] = \begin{cases} 1, & \text{if } x \text{ is true} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

#### 2.3.3 Aggregation and Averaging

When evaluating multi-label predictions that have been made for $m$ examples with $n$ labels, one has to deal with the question of how to aggregate the resulting $m \cdot n$ atomic confusion matrices. Essentially, there are four possible averaging strategies – either *(label- and example-based) micro-averaging*, *label-based (macro-)averaging*, *example-based (macro-) averaging* or *(label- and example-based) macro-averaging*. Due to the space limitations, we restrict our analysis to the most popular aggregation strategy employed in the literature, namely *micro-averaging*. This particular averaging strategy is formally defined as

$$\delta(C) = \delta\left(\sum_j \sum_i C_i^j\right) \equiv \delta\left(\sum_i \sum_j C_i^j\right), \tag{6}$$

where the $\sum$ operator denotes the cell-wise addition of confusion matrices.

## 3 Multi-Label Rule Learning

In this section, we discuss rule-based approaches for multi-label classification. We start with a brief recapitulation of inductive rule learning.

### *3.1 Rule Learning*

Rule learning has a very long history and is a well-known problem in the machine learning community (Fürnkranz et al., 2012). Over the years many different algorithms to learn a set of rules were introduced. The main advantage of rule-based classifiers is the interpretability of the models as rules can be more easily comprehended by humans than other models such as neural networks. Also, it is easy to define a syntactic generality relation, which helps to structure the search space. The structure of a rule offers the calculation of overlapping of rules as well as *more specific* and *more general*-relations. Thus, the rule set can be easily modified as opposed to most statistical models such as SVMs or neural networks. However, most rule learning algorithms are currently limited to binary or multiclass classification. Depending on the goal, one may discriminate between predictive and descriptive approaches.

#### 3.1.1 Predictive Rule Learning

*Classification rules* are commonly expressed in the form

$$\mathbf{r} : H \leftarrow B, \tag{7}$$

where the body $B$ consists of a number of *conditions*, which are typically formed from the attributes of the instance space, and the head $H$ simply assigns a value to the output attribute (e.g., $y = 0$ or $y = 1$ in binary classification). We refer to this type of rules as *single-label head* rules.

For combining such rules into predictive theories, most algorithms follow the *covering* or *separate-and-conquer* strategy (Fürnkranz, 1999), i.e., they proceed by learning one rule at a time. After adding a rule to the growing rule set, all examples covered by this rule are removed, and the next rule is learned from the remaining examples. In order to prevent overfitting, the two constraints that all examples have to be covered (*completeness*) and that negative examples must not be covered (*consistency*) can be relaxed so that some positive examples may remain uncovered and/or some negative examples may be covered by the set of rules. Typically, heuristics are used that trade off these two objectives and guide the search towards solutions that excel according to both criteria (Janssen and Fürnkranz, 2010; Minnaert et al., 2015).[1] This may also be viewed as a simple instance of the *LeGo* framework for combining local patterns (individual rules) to global theories (rule sets or decision lists) (Knobbe et al., 2008).

### 3.1.2 Descriptive Rule Learning

Contrary to predictive approaches, descriptive rule learning algorithms typically focus on individual rules. For example, *subgroup discovery* algorithms (Kralj Novak et al., 2009) aim at discovering groups of data that have an unusual class distribution, or *exceptional model mining* (Duivesteijn et al., 2016) generalizes this notion to differences with respect to data models instead of data distributions. Duivesteijn et al. (2012) extended the latter approach to MLC for finding local exceptionalities in the dependence relations between labels.

Arguably the best-known descriptive approach are *association rules* (Goethals, 2005; Hipp et al., 2000; Zhang and Zhang, 2002), which relate properties of the data in the body of the rule to other properties in the head of the rule. Thus, contrary to classification rules, where the head consists of a single class label, multiple conditions may appear in the head. Typically, association rules are found by exhaustive search, i.e., all rules that satisfy a minimum support and minimum confidence threshold are found (Agrawal et al., 1995; Zaki et al., 1997; Han et al., 2004), and subsequently filtered and/or ordered according to heuristics. Only few algorithms directly find rules that optimize a given score function (Webb, 2000). They can also be combined into theories with class association rule learning algorithms such as CBA (Liu et al., 1998, 2000; Sulzmann and Fürnkranz, 2008).

[1] Some algorithms, such as ENDER (Dembczyński et al., 2010), also find rules that directly minimize a regularized loss function.

**Table 1** Examples of different forms of multi-label rules based on the sample dataset Figure 1. Attribute names in italic denote label attributes, attributes with an overline denote negated conditions.

| Head | | Body | Example rule |
|---|---|---|---|
| Single-label | Positive | Label-independent | $quality \leftarrow$ University, Female |
| | Negative | | $\overline{tabloid} \leftarrow$ Secondary, Divorced |
| Single-label | Positive | Partially Label-dependent | $quality \leftarrow \overline{tabloid}$, University |
| | Negative | | $\overline{quality} \leftarrow tabloid$, Primary |
| Single-label | Positive | Fully Label-dependent | $\overline{sports} \leftarrow fashion$ |
| | Negative | | $\overline{sports} \leftarrow quality, tabloid$ |
| Multi-label | Partial | Label-independent | $quality, fashion \leftarrow$ University, Female |
| | Complete | | $quality, \overline{tabloid}, fashion, \overline{sports} \leftarrow$ University, Female |
| Multi-label | Partial | Partially Label-dependent | $tabloid, \overline{sports} \leftarrow fashion$, Children |
| | | Fully Label-dependent | $\overline{fashion}, \overline{sports} \leftarrow \overline{quality}, \overline{tabloid}$ |

## *3.2 Multi-Label Rules*

The goal of multi-label rule learning is to discover rules of the form

$$\mathbf{r} : \hat{\mathbf{y}} \leftarrow B \tag{8}$$

The head of the rule may be viewed as a binary prediction vector $\hat{\mathbf{y}}$, or as a set of predicted labels $\hat{P} \subset \mathscr{L}$. The body may consist of several conditions, which the examples that are covered by the rule have to satisfy. In this work, only conjunctive, propositional rules are considered, i.e., each condition compares an attribute's value to a constant by either using equality (nominal attributes) or inequalities (numerical attributes).

In mixed representation rules, labels may occur both as rule features (in the body of the rule) and as predictions (in the head of the rule). Formally, we intend to learn rules of the form

$$\mathbf{r} : y^{(i+j+k)}, \ldots, y^{(i+j+1)} \leftarrow y^{(i+j)}, \ldots, y^{(i+i)}, \phi^{(i)}, \ldots, \phi^{(1)} \tag{9}$$

in which $i \geq 0$ Boolean features, which characterize the input instances, can be mixed with $j \geq 0$ labels in the body of the rule, and are mapped to $k > 0$ different labels in the head of the rule.

Table 1 shows examples of different types of rules that can be learned from the sample dataset shown in Figure 1. One can distinguish rules according to several dimensions:

- *multi-label* vs. *single-label*: Does the head of the rule contain only a single or multiple predictions?
- *positive* vs. *negative*: Can we predict only the presence of labels or also their absence?

– *dependent* vs. *independent*: Do the predictions in part or fully depend on other labels?

The predictions in the head of the rule may also have different semantics. We differentiate between *full predictions* and *partial predictions*.

- *full predictions:* Each rule predicts a full label vector $\hat{\mathbf{y}}$, i.e., if a label $\lambda_i$ is not contained in the head, its absence is predicted, i.e., $y_i = 0$.
- *partial predictions:* Each rule predicts the presence or absence of the label only for a subset of the possible labels. For the remaining labels the rule does not make a prediction (but other rules might).

For denoting absence of labels, we will sometimes also use a bar above the labels, i.e., $\overline{\lambda}$ denotes that label $\lambda$ is predicted as non-relevant or not observed. We also allow $y = ?$ in heads $\mathbf{y}$ and $P \subset \mathscr{L} \cup \{\overline{\lambda_1}, \ldots, \overline{\lambda_n}\}$ to denote that certain labels are not concerned by a rule, i.e., that the label is neither predicted as present nor as absent.

Alternative categorizations of dependencies are possible. For example, Park and Fürnkranz (2008) categorized full label dependencies into *subset constraints* $\lambda_i \leftarrow \lambda_j$ (the instances labeled with $\lambda_j$ are a subset of those labeled with $\lambda_i$) and *exclusion constraints* $\overline{\lambda_i} \leftarrow \lambda_j$ (the instances labeled with $\lambda_i$ are disjoint from those labeled with $\lambda_j$), which can be readily expressed in a rule-based manner. Fully label dependencies are also known as *global dependencies* whereas partially label-dependent rules are also known as *local* and *semi-local dependencies*. For example, in rule (9), the features used in the body of the rule $\phi^{(1)}, \phi^{(2)}, \ldots, \phi^{(i)}$ form the local context in which the dependency $\lambda^{(i+1)}, \ldots, \lambda^{(i+j)} \rightarrow \lambda^{(i+j+1)}, \lambda^{(i+j+2)}, \ldots, \lambda^{(k)}$ holds.

### *3.3 Challenges for Multi-Label Rule Learning*

Proposing algorithms that directly learn sets of such rules is a very challenging problem, which involves several subproblems that are not or only inadequately addressed by existing rule learning algorithms.

Firstly, rule-based models expand the class of dependency models that are commonly used when learning from multi-label data. As already explained, one commonly distinguishes between *conditional* and *unconditional* label dependencies (Dembczyński et al., 2012), where the former is of a *global* nature and holds (unconditionally) in the entire instance space (regardless of any features of the instances), whereas the latter is of a *local* nature and only holds for a specific instance. By modeling semi-local dependencies that hold in a certain part of the instance space, i.e., for subgroups of the population characterized by specific features, rule-based models allow for a smooth interpolation between these two extremes. Such dependencies can be formulated elegantly via rules that mix regular features and labels in the condition part of the rule, as illustrated in Table 1. Besides, rule models offer interesting alternatives for the interpretation of dependencies. While the conventional definition

of dependency is based on probabilistic concepts, rule models are typically associated with deterministic dependencies. Yet, single rules may also be equipped with probabilistic semantics (e.g., the condition specified in the head of the rule holds with a certain probability within the region specified in the rule body).

Secondly, a rule-based formulation adds a considerable amount of flexibility to the learning process. Contrary to single-label classification, there is a large variety of loss functions according to which the performance of multi-label learning algorithms can be assessed (see Section 2.3). In a rule-based framework, a loss-minimizing head could be found for individual rules, so that the same rule body could be adapted to different target loss functions. Conversely, while conventional rule learning heuristics are targeted towards minimizing classification error aka 0/1-loss, their adaptation to different multi-target loss functions is not straightforward. Moreover, different loss functions may require different heuristics in the underlying rule learner.

Moving from learning single rules, a process which is also known as subgroup discovery, to the learning of rule sets adds another layer of complexity to the rule learning algorithms (Fürnkranz, 2005). Even an adaptation of the simple and straightforward covering strategy, which is predominantly used for learning rule sets in inductive rule learning (Fürnkranz, 1999), is a non-trivial task. For example, when learning rules with partial label heads, one has to devise strategies for dealing with examples that are partially covered, in the sense that some of their labels are covered by a rule whereas others are not. One also has to deal with possible conflicts that may arise from mixed positive and negative rules. Last but not least, one has to recognize and avoid circular dependency structures, where, e.g., the prediction of label $\lambda_i$ depends on the knowledge of a different label $\lambda_j$, which in turn depends on knowledge of $\lambda_i$. Algorithmically, we consider this the most challenging problem.

Finally, rule-based representations are directly interpretable and comprehensible to humans, at least in principle. Hence, one is able to analyze the induced rule models, including dependencies between labels discovered in the data, and may greatly benefit from the insight they provide. This is in contrast to many other types of models, for which the key information is not directly accessible. Interestingly, the possibility to inspect the algorithmic decision-making process and the right for explanation might play a major role in the up-coming European legislation (Bryce Goodman, 2016), which might even raise liability issues for manufacturers, owners and users of artificial intelligence systems.[2]

We note in passing, however, that while rules are commonly perceived to be more comprehensible than other types of hypothesis spaces that are commonly used in machine learning, the topic of learning *interpretable* rules is still not very well explored (Freitas, 2013). For example, in many studies, the comprehensibility of learned rules is assumed to be negatively correlated with their complexity, a point of view that has been questioned in recent work (Allahyari and Lavesson, 2011; Stecher et al., 2016; Fürnkranz et al., 2018). In this chapter, our main focus is on arguing that the fact that input data and labels can be used to formulate explicit

[2] *General Data Protection Regulation* 2016/679 and *Civil law rules on robotics* under the ID of 2015/2103(INL).

mixed-dependency rules has a strong potential for increasing the interpretability of multi-label learning.

## 4 Discovery of Multi-Label Rules

In this section, we review work on the problem of discovering individual multi-label rules. In Section 4.1, we discuss algorithms that are based on association rule discovery, which allow to quickly find mixed dependency rules. However, for these algorithms it often remains unclear what loss is minimized by their predictions. Other approaches, which we will discuss in Section 4.2, aim at discovering loss-minimizing rules.

### *4.1 Association Rule-Based Algorithms*

A simple, heuristic way of discovering multi-label rules is to convert the problem into an association rule discovery problem (cf. Section 3.1.2). To this end, one can use the union of labels and features as the basic itemset, discover all frequent itemsets, and derive association rules from these frequent itemsets, as most association rule discovery algorithms do. The only modification is that only rules with labels in the head are allowed, whereas potential rules with features in the head will be disregarded.

For instance, Thabtah et al. (2004) and similarly Li et al. (2008) induce single-label association rules, based on algorithms for class association rule discovery (Liu et al., 1998, 2000). Their idea is to use a multiclass, multi-label associative classification approach where single-label class association rules are merged to create multi-label rules. Allamanis et al. (2013) employ a more complex approach based on an genetic search algorithm which integrates the discovery of multi-label heads into the evolutionary process. Similarly, Arunadevi and Rajamani (2011) and Ávila et al. (2010), use evolutionary algorithms or classifier systems for evolving multi-label classification rules thereby avoiding the problem of devising search algorithms that are targeted towards that problem.

An associate multi-label rule learner with several possible labels in the head of the rules was developed by Thabtah et al. (2006). These labels are found in the whole training set, while the multi-label lazy associative approach of Veloso et al. (2007) generates the rules from the neighborhood of a test instance during prediction. The advantage then is that fewer training instances are used to compute the coverage statistics which is beneficial when small rules are a problem as they are often predicted wrong due to whole training set statistics.

A few algorithms focus on discovering global dependencies, i.e., fully label-dependent rules. Park and Fürnkranz (2008) use an association rule miner (Apriori) to discover pairwise subset (implication) and exclusion constraints, which may be

viewed as global dependencies. These are then applied in the classification process to correct predicted label rankings that violate the globally found constraints. Similarly, Charte et al. (2014) infer global dependencies between the labels in the form of association rules, and use them as a post-processor for refining the predictions of conventional multi-label learning systems. Papagiannopoulou et al. (2015) propose a method for discovering deterministic positive entailment (implication) and exclusion relationships between labels and sets of labels.

However, while all these approaches allow to quickly discover multi-label rules, it remains mostly unclear what multi-label loss the discovered rules are actually minimizing.

## *4.2 Choosing Loss-Minimizing Rule Heads*

A key advantage of rule-based methods is that learned rules can be flexibly adapted to different loss functions by choosing an appropriate head for a given rule body. Partial prediction rules, which do not predict the entire label vector, require particular attention. Very much in the same way as completeness in terms of covered examples is only important for complete rule-based theories and not so much for individual rules, completeness in terms of predicted labels is less of an issue when learning individual rules. Instead of evaluating the rule candidates with respect to only one target label, multi-label rule learning algorithms need to evaluate candidates w.r.t. all possible target variables and choose the best possible head for each candidate.

Algorithmically, the key problem is therefore to find the empirical loss minimizer of a rule, i.e., the prediction that minimizes the loss on the covered examples, i.e., we need to find the multi-label head $\mathbf{y}$ which reaches the best possible performance

$$h_{max} = \max_{\mathbf{y}} \; h(\mathbf{r}) = \max_{\mathbf{y}} \; h(\mathbf{y} \leftarrow B) \tag{10}$$

given an evaluation function $h(.)$ and a body $B$. As recently shown for the case of the F-measure, this problem is highly non-trivial for certain loss functions (Waegeman et al., 2014). Bosc et al. (2016) adapt an algorithm for subgroup discovery so that it can find the top-k multi-label rules, but the quality measure they use is based on subgroup discovery and not related to commonly used multi-label classification losses.

To illustrate the difference between measures used in association rule discovery and in multi-label rule learning, assume that the rule $\lambda_1, \lambda_2 \leftarrow B$ covers three examples $(\mathbf{x}_1, \{\lambda_2\})$, $(\mathbf{x}_2, \{\lambda_1, \lambda_2\})$ and $(\mathbf{x}_3, \{\lambda_1\})$. In conventional association rule discovery the head is considered to be satisfied for one of the three covered examples $(\mathbf{x}_2)$, yielding a precision/confidence value of $\frac{1}{3}$. This essentially corresponds to subset accuracy. On the other hand, micro-averaged precision would correspond to the fraction of 4 correctly predicted labels among 6 predictions, yielding a value of $\frac{2}{3}$.

### 4.2.1 Anti-Monotonicity and Decomposability

Rapp et al. (2018) investigate the behavior of multi-label loss functions w.r.t. two such properties, namely anti-monotonicity and decomposability. The first property which can be exploited for pruning searches—while still being able to find the best solution—is *anti-monotonicity*. This property is already well known from association rule learning (Agrawal et al., 1995; Goethals, 2005; Hipp et al., 2000) and subgroup discovery (Kralj Novak et al., 2009; Atzmüller, 2015). In the multi-label context it basically states that, if we use an anti-monotonic heuristic $h$ for evaluating rules, using adding additional labels to a head cannot improve its value if adding the previous label already decreased the heuristic value. An even stronger criterion for pruning the searches can be found particularly for decomposable multi-label evaluation measures. In few words, *decomposability* allows to find the best head by combining the single-label heads which reach the equal maximum heuristic value for a given body and set of examples. Hence, finding the best head for a decomposable heuristic comes at linear costs, as the best possible head can be deduced from considering each available label separately.

Decomposability is a stronger criterion, i.e., an evaluation measure that is decomposable is also anti-monotonic. Decomposable multi-label evaluation measures include micro-averaged rule-dependent precision, F-measure, and Hamming accuracy. Subset accuracy only fulfills the anti-monotonicity property. This can also be seen from Table 4.2.1, which shows for a large variety of evaluation measures if maximizing them for a given body can benefit from both criteria. Detailed proofs are provided by Rapp et al. (2018) and Rapp (2016).

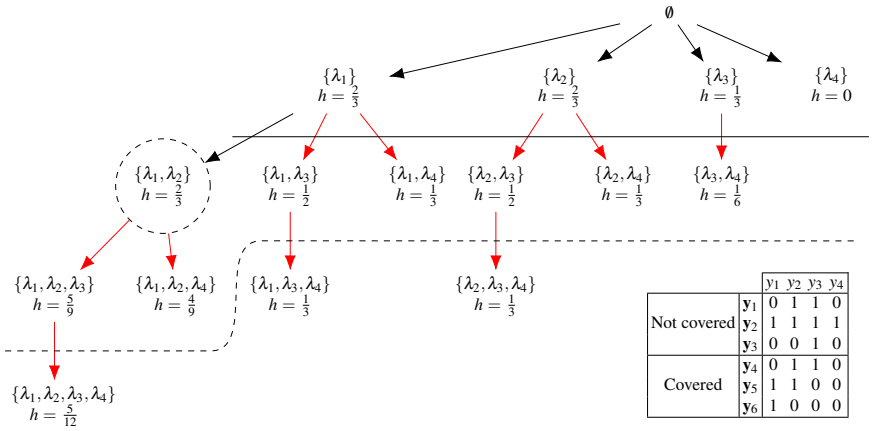
### 4.2.2 Efficient Generation of Multi-Label Heads

To find the best head for a given body different label combinations must be evaluated by calculating a score based on the used averaging and evaluation strategy. The algorithm described in the following performs a breadth-first search by recursively adding additional label attributes to the (initially empty) head and keeps track of the best rated head. Instead of performing an exhaustive search, the search space is pruned according to the findings in Section 4.1. When pruning according to anti-monotonicity unnecessary evaluations of label combinations are omitted in two ways: On the one hand, if adding a label attribute causes the performance to decrease, the recursion is not continued at deeper levels of the currently searched subtree. On the other hand, the algorithm keeps track of already evaluated or pruned heads and prevents these heads from being evaluated in later iterations. When a decomposable evaluation metric is used no deep searches through the label space must be performed. Instead, all possible single-label heads are evaluated in order to identify those that reach the highest score and merge them into one multi-label head rule.

Figure 2 illustrates how the algorithm prunes a search through the label space using anti-monotonicity and decomposability. The nodes of the given search tree

**Table 2** Anti-monotonicity and decomposability of selected evaluation functions with respect to different averaging and evaluation strategies.

| Evaluation Function | Evaluation Strategy | Averaging Strategy | Anti-Monotonicity | Decomposability |
|---|---|---|---|---|
| Precision | Partial Predictions | Micro-averaging | Yes | Yes |
| | | Label-based | Yes | Yes |
| | | Example-based | Yes | Yes |
| | | Macro-averaging | Yes | Yes |
| | Full Predictions | Micro-averaging | Yes | — |
| | | Label-based | Yes | — |
| | | Example-based | Yes | — |
| | | Macro-averaging | Yes | — |
| Recall | Partial Predictions | Micro-averaging | Yes | Yes |
| | | Label-based | Yes | Yes |
| | | Example-based | — | — |
| | | Macro-averaging | Yes | Yes |
| | Full Predictions | Micro-averaging | Yes | — |
| | | Label-based | Yes | — |
| | | Example-based | — | — |
| | | Macro-averaging | Yes | — |
| Hamming Accuracy | Partial Predictions | Micro-averaging | Yes | Yes |
| | | Label-based | Yes | Yes |
| | | Example-based | Yes | Yes |
| | | Macro-averaging | Yes | Yes |
| | Full Predictions | Micro-averaging | Yes | — |
| | | Label-based | Yes | — |
| | | Example-based | Yes | — |
| | | Macro-averaging | Yes | — |
| F-Measure | Partial Predictions | Micro-averaging | Yes | Yes |
| | | Label-based | Yes | Yes |
| | | Example-based | Yes | Yes |
| | | Macro-averaging | Yes | Yes |
| | Full Predictions | Micro-averaging | Yes | — |
| | | Label-based | Yes | — |
| | | Example-based | Yes | — |
| | | Macro-averaging | Yes | — |
| Subset Accuracy | Partial Predictions | Example-based | Yes | — |
| | Full Predictions | | — | — |




| | | $y_1$ | $y_2$ | $y_3$ | $y_4$ |
|---|---|---|---|---|---|
| Not covered | $\mathbf{y}_1$ | 0 | 1 | 1 | 0 |
| | $\mathbf{y}_2$ | 1 | 1 | 1 | 1 |
| | $\mathbf{y}_3$ | 0 | 0 | 1 | 0 |
| Covered | $\mathbf{y}_4$ | 0 | 1 | 1 | 0 |
| | $\mathbf{y}_5$ | 1 | 1 | 0 | 0 |
| | $\mathbf{y}_6$ | 1 | 0 | 0 | 0 |

**Fig. 2** Search through the label space $2^{\mathscr{L}}$ with $\mathscr{L} = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ using micro-averaged precision of partial predictions. The examples corresponding to label sets $\mathbf{y}_4, \mathbf{y}_5, \mathbf{y}_6$ are assumed to be covered, whereas those of $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$ are not. The dashed line ( - - -) indicates label combinations that can be pruned with anti-monotonicity, the solid line (—) corresponds to decomposability.

correspond to the evaluations of label combinations, resulting in heuristic values $h$. The edges correspond to adding an additional label to the head which is represented by the preceding node. As equivalent heads must not be evaluated multiple times, the tree is unbalanced.

## 5 Learning Predictive Rule-Based Multi-label Models

Predictive, rule-based theories are formed by combining individual rules into a theory. Such an aggregation step is necessary because each individual rule will only cover a part of the example space. When mixed dependency rules, i.e., rules with both labels and features in the rule bodies, are combined into a predictive theory, several problems arise that make the problem considerably harder than the aggregation of local rules into a global rule-based model (Fürnkranz, 2005).

As a very simple example, consider the case when two labels $\lambda_i$ and $\lambda_j$ always co-occur in the training data. The algorithms discussed in the previous section would then find the inclusion constraints $\lambda_i \rightarrow \lambda_j$ and $\lambda_j \rightarrow \lambda_i$. These are valid and interesting insights into the domain, but in a predictive setting, they will not help to identify both labels as positive unless at least one of the two can be predicted by another rule.[3]

[3] Similar problems have been encountered in inductive logic programming, where the learning of recursive and multi-predicate programs has received some attention (Malerba, 2003; De Raedt et al., 1993; Cameron-Jones and Quinlan, 1993).

As shown by this example, the problem of circular reasoning is a major concern in the inference with mixed dependency rules. There are two principal ways for tackling this problem. The simplest strategy is to avoid circular dependencies from the very beginning. This means that rules discovered in the learning process have to be organized in a structure that prevents cycles or, alternatively, that additional rules have to be learned with certain constraints on the set of valid conditions in the rule body.

Another way of tackling this problem is to allow for circular dependencies and generalize the inference strategy in a suitable manner. This approach has not yet received much attention in the literature. One notable exception is the work of Montañés et al. (2014) who realized this idea in so-called dependent binary relevance (DBR) learning, which is based on techniques similar to those used in conditional dependency networks (Guo and Gu, 2011).

In this section, we will describe two different approaches for tackling the first problem. One, which we call *layered learning* tries to avoid label cycles by requiring an initial guess for labels regardless of any label dependence (Section 5.1).While this approach is rather coarse in that batches of rules are learned, we will then also consider approaches that try to adapt the covering or separate-and-conquer strategy, which is frequently used in inductive rule learning (Section 5.2).

## *5.1 Layered Multi-label Learning*

The recently very popular classifier chains (CC; Read et al., 2011) were found to be an effective approach for avoiding label cycles. Their key idea is to use an arbitrary order $\lambda_1, \lambda_2, \ldots, \lambda_n$ on the labels, and learn rules that involve predictions for $\lambda_i$ only from the input attributes and all labels $\lambda_j, j < i$. This, however, has some obvious disadvantages, which have been addressed by many variants that have been investigated in the literature.

One drawback is the (typically randomly chosen) predetermined, fixed order of the classifiers (and hence the labels) in the chain, which makes it impossible to learn dependencies in the opposite direction. This was already recognized by Malerba et al. (1997), who built up a very similar system in order to learn multiple dependent concepts. In this case, the chain on the labels was determined beforehand by a statistical analysis of the label dependencies. Still, using a rule learner for solving the resulting binary problems would only allow to induce rules between two labels in one direction.

### 5.1.1 Stacked Binary Relevance

An alternative approach without this limitation is to use two levels of classifiers: the first one tries to predict labels independently of each other, whereas the second level of classifiers makes additional use of the predictions of the previous level. More

| | |
|---|---|
| *quality* ← University | $\overline{\textit{tabloid}}$ ← Primary, Single |
| $\overline{\textit{quality}}$ ← Single | $\overline{\textit{tabloid}}$ ← Female, Married |
| *quality* ← Female | $\overline{\textit{tabloid}}$ ← Divorced, Secondary, $\overline{\text{Children}}$ |
| *quality* ← Secondary | *tabloid* ← true |
| $\overline{\textit{quality}}$ ← true | — |
| — | $\overline{\textit{tabloid}}$ ← Primary, Single |
| *quality* ← University | *tabloid* ← $\overline{\textit{quality}}$ ← true |
| *quality* ← *sports*, Secondary | $\overline{\textit{tabloid}}$ ← Female, Married |
| $\overline{\textit{quality}}$ ← Male | $\overline{\textit{tabloid}}$ ← Secondary, Divorced |
| $\overline{\textit{quality}}$ ← true | *tabloid* ← true |

| | |
|---|---|
| $\overline{\textit{fashion}}$ ← Male | *sports* ← Divorced, Secondary, $\overline{\text{Children}}$ |
| $\overline{\textit{fashion}}$ ← $\overline{\text{Children}}$ | *sports* ← Children, Male |
| *fashion* ← true | $\overline{\textit{sports}}$ ← true |
| — | — |
| $\overline{\textit{fashion}}$ ← Male | $\overline{\textit{sports}}$ ← $\overline{\textit{quality}}$, *tabloid* |
| $\overline{\textit{fashion}}$ ← $\overline{\text{Children}}$, $\overline{\textit{tabloid}}$ ← | *sports* ← Secondary |
| *fashion* ← true | *sports* ← Children , Male |
| | $\overline{\textit{sports}}$ ← true |

**Fig. 3** Rule set obtained from layered learning on the example dataset (Figure 2). Decision lists from first and second level are separated by —.

specifically, the training instances for the second level are expanded by the label information of the other labels, i.e., a training example $\mathbf{x}$ for label $y_i$ is transformed into $(x_1,\ldots,y_1,\ldots,y_{i-1},y_{i+1},\ldots,y_n)$. During training, the prediction of the first level of classifiers is used as additional features for the second level, i.e., the final prediction $\hat{y}_j$ depends on predictions $f_i(\mathbf{x})$ and $f'_j(\mathbf{x},f_1(\mathbf{x}),\ldots,f_n(\mathbf{x}))$. Hence, each label will be characterized by two sets of rule models, namely the rules $R_i$ which depend only on instance features, and a second set of rule models $R_i^*$ depending (possibly) also on other labels. $R_i$ can then provide the predictions that are necessary for executing the rules in $R_i^*$. Loza Mencía and Janssen (2014, 2016) refer to this technique as *stacked binary relevance* (SBR) in contrast to plain, unstacked binary relevance learning.

Figure 3 shows a rule set that can be obtained with SBR for the sample dataset of Figure 2. One can see that two separate decision lists are learned for each label using a conventional rule learner such as *Ripper* Cohen (1995). The top list $R_i$ is learned only from the input features, and the bottom part $R_i^*$ is learned from input features and the predictions originating from the top rule set.

Despite being able to learn, in contrast to CC, relationships in either direction and in any constellation, this method still has its shortcomings. Firstly, it requires comprehensive predictive rules for each label on the first level even though the labels may effectively be predicted based on other labels. For example, assume the global relation $\lambda_i \leftarrow \lambda_j$, the approach would need to learn a rule model $R_j$ once for predicting $\lambda_j$ and once implicitly as part of $R_i$.

Secondly, a limitation of the stacking approach may appear when circular dependencies exist between labels. A very simple example is if two labels exclude each

other, i.e., if both relationships $\lambda_i \leftarrow \overline{\lambda_j}$, and $\lambda_j \leftarrow \overline{\lambda_i}$ hold.Such rules could lead to contradictions and inconsistent states. For instance, assume that both labels $\lambda_i$ and $\lambda_j$ were predicted as relevant at the first layer. Both predictions would be flipped at the next layer, leading again to an (irresolvable) inconsistency according to the learned rules.

Another problem that needs to be addressed by layered algorithms is the problem of *error propagation*: If the prediction for a label $\lambda_j$ depends on another label $\lambda_i$, a mistake on the latter is likely to imply a mistake on the former (Senge et al., 2012).

Finally and most notably, the method of (Loza Mencía and Janssen, 2016) is limited to produce single-label head rules.

Several variants were proposed in the literature, which in deviation from the basic technique may use predictions instead of the true label information as input (Montañés et al., 2014), only the predictions in a pure stacking manner (Godbole and Sarawagi, 2004), or Gibbs sampling instead of the first level of classifiers (Guo and Gu, 2011). *Dependent binary relevance* (Montañés et al., 2014) and *conditional dependency networks* (Guo and Gu, 2011) are particularly concerned with estimating probability distributions (especially joint distribution). They both use logistic regression as their base classifier, which is particularly adequate for estimating probabilities. This type of models are obviously much harder to comprehend than rules, especially for higher number of input features. Therefore, the label dependencies would remain hidden somewhere in the model, even though they may have been taken into account and accurate classifiers may have been obtained. A more general approach is to integrate the stacking of label features directly into the covering loop. Adaptations of the separate-and-conquer strategy to the multi-label case will be discussed in the next section.

## *5.2 Multi-Label Separate-and-Conquer*

The most frequently used strategy for learning a rule-based predictive theory is the so-called *covering* or *separate-and-conquer* strategy, which is either integrated into the rule learning algorithm (Fürnkranz, 1999) or used as a post-processor for selecting a suitable subset among previously learned rules (Liu et al., 1998). Although it may seem quite straightforward, its adaptation to the multi-label case is only trivial if complete rules are learned, i.e., if each rule predicts a complete assignment of the label vector. In this case, one may learn a decision list with the covering strategy, which removes all examples that are covered by previous rules before subsequent rules are learned. In this way, the learning strategy essentially mirrors the sequential nature in which predictions are made with a decision list. In the context of multi-label classification, this strategy corresponds to applying the well known *label powerset* transformation which converts each label combination in the data into a meta-label and then solves the resulting multiclass problem (cf. Tsoumakas et al., 2010).

**Require:** New training example pairs $\mathscr{T} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\}$
1: $\mathscr{T} = \{(\mathbf{x}_1, \hat{\mathbf{y}}_1), \ldots, (\mathbf{x}_m, \hat{\mathbf{y}}_m)\}$ with $\hat{\mathbf{y}}_i = (?, ?, \ldots, ?)$, $i = 1 \ldots m$
2: **while** $\mathscr{T}$ not empty **do**
3: $\quad \mathbf{r} \leftarrow$ *findBestGlobalRule*$(\mathscr{T})$ ▷ find best rule by refining rule body (and head) w.r.t. some heuristic $h$
4: $\quad$ apply $\mathbf{r}$: apply header on covered $\mathbf{x}_i \in \mathscr{T}$ and put them into $\mathscr{T}_{cov}$
5: $\quad$ **if** enough $\mathbf{x}_i$ in $\mathscr{T}_{cov}$ with fully covered labels, i.e., $\forall j.\ (\hat{\mathbf{y}}_i)_j \neq ?$, **then**
6: $\quad\quad$ make $\mathbf{r}$ full prediction rule and do not add $\mathscr{T}_{cov}$ to $\mathscr{T}$
7: $\quad$ **else**
8: $\quad\quad$ re-add $\mathscr{T}_{cov}$ to $\mathscr{T}$
9: $\quad$ add $\mathbf{r}$ to decision list $R$
10: **return** decision list $R$

**Fig. 4** General training algorithm for the multi-label separate-and-conquer algorithm.

However, in the case of partial-prediction rules, the situation becomes considerably more complex. One can, e.g., not simply remove all examples that are covered by a rule, because the rule will in general only predict a subset of the relevant labels. An alternative strategy might be to remove all predicted *labels* from the examples that are covered by the rule: if a rule $\mathbf{r} : \hat{\mathbf{y}} \leftarrow B$ covers a training example $(\mathbf{x}, \mathbf{y})$, the example is not removed but replaced with the example $(\mathbf{x}, \mathbf{y} \setminus \hat{\mathbf{y}})$. In this way, each training example remains in the training set until all of its labels are covered by at least one rule. However, even this strategy may be problematic, because removing labels from covered training instances in this way may distort the label dependencies in the training data.

By using separate-and-conquer strategy to induce a rule model, two of the shortcomings of the layered approach from the previous section are addressed. Firstly, the iterative, non-parallel induction of rules in the covering process ensures that redundant rules are avoided because of the separation step. Secondly, cyclic dependencies cannot longer harm the induction or prediction process since the order in which labels are covered or predicted is naturally chosen by the covering process. Similarly, the learner may also dynamically model local label dependencies and does not depend on a global order as in classifier chains.

### 5.2.1 A Multi-Label Covering Algorithm

Figure 4 shows the multi-label covering algorithm proposed by Loza Mencía and Janssen (2016). The algorithm essentially proceeds as sketched described above, i.e., covered examples are not removed entirely but only the subset of predicted labels is deleted from the example.

For learning a new multi-label rule (line 3), the algorithm performs a top-down greedy search, starting with the most general rule. By adding conditions to the rule's body it can successively be specialized, resulting in fewer examples being covered. Potential conditions result from the values of nominal attributes or from averaging two adjacent values of the sorted examples in case of numerical attributes. When-

| | |
|---|---|
| $\overline{fashion} \leftarrow$ Male | $\overline{tabloid} \leftarrow$ Primary |
| $\overline{sports} \leftarrow$ Children | $tabloid \leftarrow$ Single |
| $\overline{quality} \leftarrow$ Primary | $\overline{tabloid} \leftarrow$ Secondary |
| $quality \leftarrow$ University | $\overline{sports} \leftarrow$ true |
| $\overline{sports} \leftarrow \overline{quality} \leftarrow$ true | $tabloid \leftarrow$ Divorced |
| $\overline{tabloid} \leftarrow$ Female, $\overline{\text{Children}}$ | $\overline{tabloid} \leftarrow$ true |
| $fashion \leftarrow$ Children | $\overline{fashion} \leftarrow$ Married |
| $\overline{sports} \leftarrow$ University | $\overline{fashion}, * \leftarrow sports$ |
| $\overline{quality} \leftarrow$ Single | $fashion \leftarrow$ true |
| $quality \leftarrow \overline{\text{Children}}$ | $\overline{quality}, * \leftarrow tabloid$ |
| $sports \leftarrow$ Divorced | $\overline{quality}, * \leftarrow$ true |
| $tabloid \leftarrow$ Married, Male | |

(a) Single-label head rules (read column-wise)

| |
|---|
| $\overline{quality}, \overline{fashion}, \overline{sports} \leftarrow$ Primary |
| $quality, \overline{sports} \leftarrow$ University |
| $\overline{fashion} \leftarrow$ Male |
| $\overline{quality}, tabloid, fashion, \overline{sports} \leftarrow$ Single, Secondary |
| $quality, \overline{tabloid}, sports \leftarrow$ Female |
| $quality, tabloid, fashion, \overline{sports} \leftarrow$ Married |
| $quality, \overline{tabloid}, \overline{fashion}, sports \leftarrow$ Secondary, $\overline{\text{Children}}$ |
| $\overline{quality}, fashion, \overline{sports} \leftarrow$ true |
| $\overline{tabloid}, * \leftarrow \overline{quality} \leftarrow$ true |
| $tabloid, * \leftarrow$ true |

(b) Multi-label head rules

**Fig. 5** Decision lists induced from the sample dataset of Figure 2 with precision as heuristic. The stars (*) indicate full prediction rules, after which the prediction stops if the rule fires.

ever a new condition is added, a corresponding single- or multi-label head that predicts the labels of the covered examples as accurate as possible must be found (cf. Section 4.2 and, in particular, Figure 2).

If a new rule is found, the predicted labels from the examples are marked as covered by these rules, i.e., $(\mathbf{y}_i)_j$ are set to 0 or 1, respectively. As depicted in lines 5–6 in the pseudo-code of Figure 4, only examples for which enough labels have been predicted can be entirely removed from the training set. A rule that predicts many of such examples is marked as full prediction rule, which means that the execution of the decision list may stop after this rule has fired.

To classify test examples, the learned rules are applied in the order of their induction. If a rule fires, the labels in its head are applied unless they were already set by a previous rule.[4] The process continues with the next rule in the multi-label decision list until either a specially marked full prediction rule is encountered or all rules of the decision list have been processed.

[4] This corresponds to the default strategy in classification rule learning, where rules are appended at the end of a list. Note however, that there are also good arguments for prepending rules at the beginning of the list, so that, e.g., exceptions are handled before the general rule (Webb, 1994).

**Table 3** Statistics of the used datasets: name of the dataset, domain of the input instances, number of instances, number of nominal/binary and numeric features, total number of unique labels, average number of labels per instance (cardinality), average percentage of relevant labels (label density), number of distinct label sets in the data.

| Name | Domain | Instances | Nominal | Numeric | Labels | Cardinality | Density | Distinct |
|---|---|---|---|---|---|---|---|---|
| EMOTIONS | music | 593 | 0 | 72 | 6 | 1.869 | 0.311 | 27 |
| SCENE | image | 2407 | 0 | 294 | 6 | 1.074 | 0.179 | 15 |
| FLAGS | image | 194 | 9 | 10 | 7 | 3.392 | 0.485 | 54 |
| YEAST | biology | 2417 | 0 | 103 | 14 | 4.237 | 0.303 | 198 |
| BIRDS | audio | 645 | 2 | 258 | 19 | 1.014 | 0.053 | 133 |
| GENBASE | biology | 662 | 1186 | 0 | 27 | 1.252 | 0.046 | 32 |
| MEDICAL | text | 978 | 1449 | 0 | 45 | 1.245 | 0.028 | 94 |
| ENRON | text | 1702 | 1001 | 0 | 53 | 3.378 | 0.064 | 753 |
| CAL500 | music | 502 | 0 | 68 | 174 | 26.0 | 0.150 | 502 |

Note that if we had only a single binary (or multiclass) label, i.e. $n = 1$, the described algorithm would behave exactly as the original separate-and-conquer approach. However, for $n > 1$ the algorithm re-adds partially and even fully covered examples instead of removing them (line 8). These examples may serve as an anchor point for subsequent rules and facilitate in such a manner the rule induction process. Moreover, this step enables the algorithm to induce rules which test for the presence of labels. These type of rules are of particular interest since they explicitly reveal label dependencies discovered in the dataset.

Figure 5 shows the results of applying these algorithms to our sample dataset. The top part shows the rules obtained with the single-label head version of Loza Mencía and Janssen (2016), whereas the lower part shows those of the multi-label head extension by Rapp et al. (2018).

## 6 Case Studies

In this section, we show a few sample result obtained with some of the algorithms described in the previous sections on commonly used benchmark data. Our main focus lies on the inspection and the analysis of the induced rule models, and not so much on their predictive accuracy in comparison to state-of-the-art multi-label classification methods (generally, the predictive performance of rule-based models will be lower). We primarily show some sample rule models, but also discuss statistics on the revealed dependencies.

We experimented with several datasets from the MULAN repository.[5] Table 3 gives a brief overview of the used datasets, along with characteristics such as the number of instances, the number and nature of the attributes, as well as some characteristics on the distribution of labels. The datasets are from different domains and have varying properties. Details of the data are given in the analysis when needed.

[5] http://mulan.sf.net/datasets.html

| **Approach** | YEAST | |
|---|---|---|
| *BR* | *Class4* ← x23 > 0.08, x49 < -0.09<br>*Class4* ← x68 < 0.05, x33 > 0.00, x24 > 0.00, x66 > 0.00, x88 > -0.06<br>*Class4* ← x3 < -0.03, x71 > 0.03, x91 > -0.01<br>*Class4* ← x68 < 0.03, x83 > -0.00, x44> 0.029, x93 < 0.01<br>*Class4* ← x96 < -0.03, x10 > 0.01, x78< -0.07 | |
| *SBR* | *Class4* ← *Class3*, $\overline{Class2}$<br>*Class4* ← *Class5*, $\overline{Class6}$<br>*Class4* ← *Class3*, *Class1*, x22 > -0.02 | |
| *Single* | *Class4* ← *Class3*, x91 > -0.02, x50 < -0.02, x68 < 0.03<br>*Class4* ← *Class3*, x90 > -0.02, x77 < -0.04<br>*Class4* ← x60 < -0.03, x57 < -0.07, x19 > -0.01 | |
| | MEDICAL | ENRON |
| *BR* | *Cough* ← “cough”, $\overline{\text{“lobe”}}$<br>*Cough* ← “cough”, “atelectasis”<br>*Cough* ← “cough”, “opacity”<br>*Cough* ← “cough”, “airways”<br>*Cough* ← “cough” , $\overline{\text{“pneumonia”}}$, $\overline{\text{“2”}}$<br>*Cough* ← “coughing”<br>*Cough* ← “cough”, “early” | *Joke* ← “mail”, “fw”, ”didn” |
| *SBR* | *Cough* ← “cough” , $\overline{Pneumonia}$ , $\overline{Pulmonary\_collapse}$ , $\overline{Asthma}$<br>*Cough* ← “coughing”<br>*Cough* ← *Asthma*, “mild” | *Joke* ← *Personal*, “day”, “mail” |
| *Single* | *Cough* ← “cough”, $\overline{\text{“lobe”}}$, $\overline{\text{“asthma”}}$<br>*Cough* ← “cough”, “opacity”<br>*Cough* ← “cough”, “atelectasis”<br>*Cough* ← “cough”, “airways”<br>*Cough* ← “cough”, *Fever* | *Joke* ← “didn”, $\overline{\text{“wednesday”}}$<br>*Joke* ← *Personal*, “forwarded” |

**Fig. 6** Example rule sets for one exemplary label, respectively, learned by BR, SBR and separate-and-conquer (single).

## *6.1 Case Study 1: Single-Label Head Rules*

In the first case study, we compared several single-head multi-label rule learning algorithms, namely conventional binary relevance (*BR*), the layered algorithm stacked binary relevance (*SBR*), and a separate-and-conquer learner seeking for rules with only a single label in the head (*Single*). The rule learner *Ripper* (Cohen, 1995) was used for finding the label-specific candidate single-head candidate rules. Among these, the best was selected according to the micro-averaged F-measure (Loza Mencía and Janssen, 2016).

In the following, we first take a closer look on the actual rules, comparing them to the rules induced separately for each label and by separate-and-conquer. Subsequently, we put the focus on visualizing dependencies between labels found by the stacking approach. We refer to Loza Mencía and Janssen (2016) for extensive statistics and more detailed evaluations.

### 6.1.1 Exemplary Rule Models

Examples of learned rule sets are shown in Figure 6. In the case of YEAST, we see a much more compact and less complex rule set for *Class4* for the layered learner *SBR* than for the independently learned *BR* classifier. The rule set also seems more appropriate for a domain expert to understand coherences between proteins (instance features) and protein functions (labels). The separate-and-conquer model *Single* is less explicit in this sense, but it shows that certainly *Class3* is an important class for expressing *Class4*.[6]

Figure 6 also shows the models for the diagnosis *Cough* in the MEDICAL task. This dataset is concerned with the assignment of international diseases codes (ICD) to real, free-text radiological reports. Interestingly, the model found by *SBR* reads very well, and the found relationship seems to be even comprehensible to non-experts. For example, the first rule can be read as

> *If the patient does not have* Pneumonia*, a* Pulmonary_collapse *or* Asthma *and "cough"s or is "coughing", he just has a* Cough*. Otherwise, he may also have a "mild"* Asthma*, in which case he is also considered to have a* Cough*.*

The theory learned by *Single* is quite similar to the one learned by simple *BR*, which shows that the textual rules were considered to be more powerful than the dependency-based rule. Only at the end, a local dependency is learned: *Cough* only depends on the word "cough" if the label for *Fever* has also been set.

In ENRON, which is concerned with the categorization of emails during the Enron scandal, the learned models are generally less comprehensible. The observed relation between *Personal* and *Joke* can clearly be explained from the hierarchical structure on the topics.

Regarding the sizes of the models, we found between 50 and 100 rules for YEAST and MEDICAL, and between 76 (*BR*) and 340 (*Single*) for ENRON. Note, however, that even for ENRON this results in an average of only 6.4 rules per label for the largest rule model. Moreover, only a fraction of them are necessary in order to track and comprehend the prediction for a particular test instance. For instance, the report

> *Clinical history: Cough for one month.*
> *Impression: Mild hyperinflation can be seen in viral illness or reactive airway disease. Streaky opacity at the right base is favored to represent atelectasis.*

in the MEDICAL dataset was classified by experts as normal *Cough*, as well as by the rule sets in Figure 6. Furthermore, the rule models allow to identify the relationships found by the algorithm responsible for the prediction and even the training examples responsible for finding such patterns. This type of inspection may facilitate in a more convenient way than with black box approaches the integration of expert feedback—for instance on the semantic coherence (Gabriel et al., 2014) or plausibility of rules—and also an interaction with the user. For example, Beckerle (2009) explored an interactive rule learning process where learned rules could be directly modified by the user, thereby causing the learner to re-learn subsequently learned rules.

[6] For convenience, we only show the rules with this label in the head.

### 6.1.2 Visualization of Dependencies

Figure 7 shows a graphical representation of the label-dependent rules found by *SBR* on some of the smaller datasets. Each graph shows a correlation between label pairs. Labels are enumerated from 1 to the number of labels, and the corresponding label names are shown at the bottom of the coordinate system. Blue boxes in the intersection square between a *row label* and a *column label* depict fully label-dependent rules, green boxes show partially label-dependent rules. A colored box at the top corners indicates a rule of the type

$$\textit{row label} \leftarrow \ldots, \textit{column label}, \ldots,$$

whereas the bottom corners represent the opposite

$$\textit{column label} \leftarrow \ldots, \textit{row label}, \ldots$$

rules. As an example, the blue box in the upper left corner of the square in the second row and fourth column in Figure 7a (EMOTIONS) indicates that the algorithm found a rule of the type

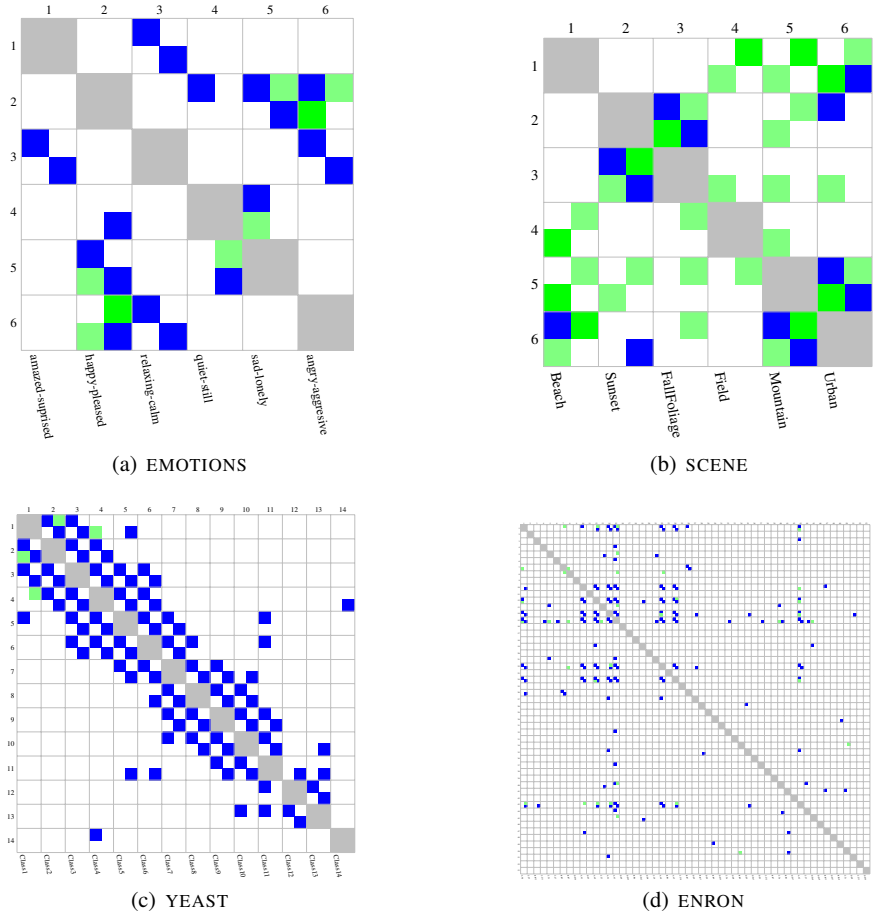
$$\textit{happy-pleased} \leftarrow \ldots, \text{quiet-still}, \ldots,$$

i.e., that quiet or still audio sample cause (possibly together with other factors) happy or pleased emotions. Note, however, that the graphs do not show whether the head or conditions are positive or negative.

In particular in SCENE (Figure 7b), we find many local dependencies, which also depend on some instance features. This is reasonable, since the task in this dataset is to predict elements of a scenery image, and although some label combinations may be more likely than others, whether an element is present or not will still depend on the content of the picture at hand. In YEAST the labels seem to be organized in a special way since we encounter the pattern that a label depends on its preceding and the two succeeding labels. ENRON has a hierarchical structure on its labels, which can be recognized from the vertical and horizontal patterns originating from parent labels.

### 6.1.3 Discussion

In our experiments, a layered learning approach such as *SBR* proved to be particularly effective at inducing rules with labels as conditions in the bodies of the rules. The resulting models turned out to be indeed very useful for discovering interesting aspects of the data, which a conventional single-label rule learner is unable to uncover. The visualizations shown above also confirm that numerous explicit local and global dependencies can be found in these database. However, we also found that the GENBASE dataset exhibits only very weak label dependencies, which can hardly be exploited in order to improve the predictive performance, despite the fact that this dataset is frequently used for evaluating multi-label algorithms.




(a) EMOTIONS (b) SCENE

(c) YEAST (d) ENRON

**Fig. 7** Visualization of the label-dependent rules for *SBR*. Rows and columns correspond to labels, green entries represent local dependencies and blue entries global dependencies that involve the corresponding label pairs (more details in the text).

## *6.2 Case Study 2: Multi-Label Heads*

The second case study compares *BR*, *Single* and *Multi* for candidate rule selection (Rapp et al., 2018).[7] Its main purpose was to demonstrate the applicability of a covering approach for inducing multi-label head rules despite the exponentially large search space.

[7] The source code of the employed algorithms and more extensive evaluations are available at https://github.com/keelm/SeCo-MLC

| | |
|---|---|
| *red*, *green*, *blue*, *yellow*, *white* ← colors>5, stripes≤3 | (65,0) |
| $\overline{red}$, *green*, $\overline{blue}$, *yellow*, *white*, $\overline{black}$, $\overline{orange}$ ← animate, stripes≤0, crosses≤0 | (11,0) |

| | | | | |
|---|---|---|---|---|
| *yellow* ← colors>4 | (21,0) | *green* ← text | (11,0) |  |
| *red* ← *yellow* | (21,0) | $\overline{orange}$ ← saltires<1 | (1,0) | |
| *blue* ← colors>5 | (14,0) | $\overline{black}$ ← area<11 | (12,0) | |
| *white* ← *blue* | (14,0) | | | |

**Fig. 8** Example of learned multi- and single-label head rule lists learned in the FLAGS dataset. In parentheses, we show $(TP,FP)$, the number of positive and negative examples covered by each rule. Shown are all rules that cover the flag of the US Virgin Islands, which is shown in the lower right corner.

### 6.2.1 Exemplary Rule Models

The extended expressiveness of multi-label head rules can be illustrated by the rules shown in Figure 8 that have been learned on the data set FLAGS, which maps characteristics of a flag and its corresponding country to the colors appearing on the flag. Shown are all rules that concern the flag of the US Virgin Islands, which is also shown in the table. Whereas in this case the single-label heads allow an easier visualization of the pairwise dependencies between characteristics/labels and labels, the multi-label head rules allow to represent more complex relationships and provide a more direct explanation of why the respective colors are predicted for the flag. Note that the rules form decision lists, which are applied in order until all labels are set, and later rules cannot overwrite earlier rules. Thus the first rule sets the colors *red*, *green*, *blue*, *yellow*, and *white*, whereas the second rule determines that *black* and *orange* do not occur. The other labels are already set by the previous rule and are not overwritten. No further rules would be considered for the prediction because all labels are already assigned.

This example also illustrates that while decision lists are conceptually easy to extend to the multi-label case by removing covered labels, the interpretability of the resulting rules may suffer. Learning rule sets that collectively determine the predicted label set from multiple possibly overlapping or contradicting partial predictions is an open question for future work.

Whether more labels in the head are more desirable or not highly depends on the data set at hand, the particular scenario and the preferences of the user, as generally do comprehensibility and interpretability of rules. These issues cannot be solved by the presented methods. However, the flexibility of being able to efficiently find loss-minimizing multi-label heads for a variety of loss functions can lay the foundation to further improvements, gaining better control over the characteristics of the induced model and hence better adaption to the requirements of a particular use case.

When analyzing the general characteristics of the models which have been learned by the proposed algorithm, it becomes apparent that multi-label head rules are particularly learned when using the precision metric, rather than one of the other metrics. The reason is that precision only considers the covered examples whereas for the other metrics the performance also depends on uncovered examples. Hence,

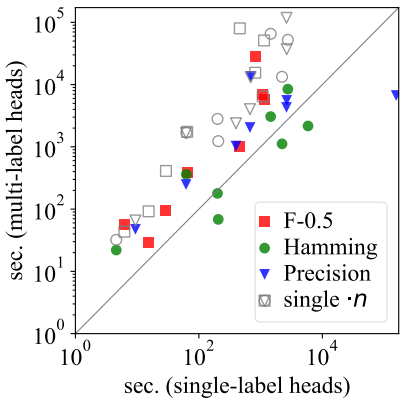



**Fig. 9** Training times for the separate-and-conquer algorithm. Direct comparison between learning single-label and multi-label heads.

it is very likely that the performance of a rule slightly decreases when adding an additional label to its head, which in turn causes single-label heads to be preferred.[8]

### 6.2.2 Predictive Performance

Because of this bias towards single-label rules for most of the metrics, large differences in predictive performance of single-label and multi-label head decision lists cannot be expected. We therefore only summarize the main finding, which compared the algorithms' performance using a Friedman test (Friedman, 1937) and a Nemenyi post-hoc test (Nemenyi, 1963) following the methodology described by Demšar (2006). The null hypothesis of the Friedman test ($\alpha = 0.05, N = 8, k = 10$) that all 10 algorithms have the same predictive quality on the eight datasets shown in Figure 3 (excluding ENRON) could not be rejected for many of the evaluation measures, such as subset accuracy and micro- and macro-averaged F1. In the other cases, the Nemenyi post-hoc test was not able to assess a statistical difference between different algorithms that used the same objective for optimizing the rules.

### 6.2.3 Computational Cost

Figure 9 shows the relation between the time spent for finding single- vs. multi-label head rules using the same objective and data set. The empty forms denote the single-label times multiplied by the number of labels in the data set and represent an approach with a computational complexity increased by one polynomial order w.r.t. number of labels. Note that full exploration of the labels space was al-

[8] The inclusion of a factor which takes the head's size in account could resolve this bias and lead to heads with more labels, but this is subject to future work.

ready intractable for the smaller data sets on our system, and became only feasible through the use of anti-monotonicity and decomposability pruning, as described in Section 4.2. We can observe that the costs for learning multi-label head rules are in the same order of magnitude as the costs for learning single-label head rules, despite the need for exploring the full label space for each candidate body.

## 7 Conclusion

In this work, we recapitulated recent work on inductive rule learning for multi-label classification problems. The main advantage of such an approach is that mixed dependencies between input variables and labels can be seamlessly integrated into a single coherent representation, which facilitates the interpretability of the learned multi-label models. However, we have also seen that combining multi-label rules into interpretable predictive theories faces several problems, which are not yet sufficiently well addressed by current solutions. One problem is that mixed-dependency rules needs to be structured in a way that allows each label that occurs in the body of a rule to be predicted by some other rule in a way that avoids cyclic reasoning. We have seen two principal approaches to solve this problem, a layered technique that relies on a pre-defined structure of the prediction and rule induction process, and a second approach that relies on adapting the separate-and-conquer or covering strategy from single-label rule learning to the multi-label case. The results we have shown in several domains are encouraging, but it is also clear that they are still somewhat limited. For example, the multi-label decision lists that result from the latter approach are hard to interpret because of the implicit dependencies that are captured in the sequential interpretation of the rules. Thus, multi-label rule learning remains an interesting research goal, which combines challenging algorithmic problems with a strong application potential.

**Acknowledgements** We would like to thank Frederik Janssen for his contributions to this work. Computations for this research were conducted on the Lichtenberg high performance computer of the TU Darmstadt.

## References

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 307–328. AAAI Press (1995)

Allahyari, H., Lavesson, N.: User-oriented assessment of classification model understandability. In: Kofod-Petersen, A., Heintz, F., Langseth, H. (eds.) Proceedings of the 11th Scandinavian Conference on Artificial Intelligence (SCAI-11). Frontiers in Artificial Intelligence and Applications, vol. 227, pp. 11–19. IOS Press, Trondheim, Norway (2011)

Allamanis, M., Tzima, F., Mitkas, P.: Effective Rule-Based Multi-label Classification with Learning Classifier Systems. In: Adaptive and Natural Computing Algorithms, 11th International Conference, ICANNGA 2013. pp. 466–476 (2013)

Arunadevi, J., Rajamani, V.: An evolutionary multi label classification using associative rule mining for spatial preferences. IJCA Special Issue on Artificial Intelligence Techniques - Novel Approaches and Practical Applications (3), 28–37 (2011)

Atzmüller, M.: Subgroup discovery. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 5(1), 35–49 (2015)

Ávila, J., Galindo, E., Ventura, S.: Evolving Multi-label Classification Rules with Gene Expression Programming: A Preliminary Study. In: Hybrid Artificial Intelligence Systems. vol. 6077, pp. 9–16. Springer (2010)

Beckerle, M.: Interaktives Regellernen. Diploma thesis, Technische Universtität Darmstadt (2009), in German

Bosc, G., Golebiowski, J., Bensafi, M., Robardet, C., Plantevit, M., Boulicaut, J.F., Kaytoue, M.: Local subgroup discovery for eliciting and understanding new structure-odor relationships. In: Calders, T., Ceci, M., Malerba, D. (eds.) Proceedings of the 19th International Conference on Discovery Science (DS-16). Lecture Notes in Computer Science, vol. 9956, pp. 19–34. Bari, Italy (2016)

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

Bryce Goodman, S.F.: European union regulations on algorithmic decision-making and a "right to explanation". In: Proceedings of the 2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016). pp. 26–30 (2016), arXiv:1606.08813 [stat.ML]

Cameron-Jones, R.M., Quinlan, J.R.: Avoiding pitfalls when learning recursive theories. In: Bajcsy, R. (ed.) Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93). pp. 1050–1057. Chambéry, France (1993)

Charte, F., Rivera, A.J., del Jesús, M.J., Herrera, F.: LI-MLC: A label inference methodology for addressing high dimensionality in the label space for multilabel classification. IEEE Transactions on Neural Networks and Learning Systems 25(10), 1842–1854 (2014)

Chekina, L., Gutfreund, D., Kontorovich, A., Rokach, L., Shapira, B.: Exploiting label dependencies for improved sample complexity. Machine Learning 91(1), 1–42 (2013)

Cohen, W.W.: Fast effective rule induction. In: Prieditis, A., Russell, S. (eds.) Proceedings of the 12th International Conference on Machine Learning (ML-95). pp. 115–123. Morgan Kaufmann, Lake Tahoe, CA (1995)

De Raedt, L., Lavrač, N., Džeroski, S.: Multiple predicate learning. In: Bajcsy, R. (ed.) Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93). pp. 1037–1043. Morgan Kaufmann, Chambéry, France (1993)

Dembczyński, K., Kotłowski, W., Słowiski, R.: ENDER: a statistical framework for boosting decision rules. Data Mining and Knowledge Discovery 21(1), 52–90 (2010)

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

Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)

Duivesteijn, W., Feelders, A., Knobbe, A.J.: Exceptional model mining – supervised descriptive local pattern mining with complex target concepts. Data Mining and Knowledge Discovery 30(1), 47–98 (2016)

Duivesteijn, W., Loza Mencía, E., Fürnkranz, J., Knobbe, A.J.: Multi-label lego – enhancing multi-label classifiers with local patterns. In: Hollmén, J., Klawonn, F., Tucker, A. (eds.) Advances in Intelligent Data Analysis XI – Proceedings of the 11th International Symposium on Data Analysis (IDA-11). Lecture Notes in Computer Science, vol. 7619, pp. 114–125. Springer (2012)

Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) Advances in Neural Information Processing Systems. vol. 14, pp. 681–687. MIT Press (2001)

Freitas, A.A.: Comprehensible classification models: a position paper. SIGKDD Explorations 15(1), 1–10 (2013)

Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association 32, 675–701 (1937)

Fürnkranz, J.: Separate-and-conquer rule learning. Artificial Intelligence Review 13(1), 3–54 (1999)

Fürnkranz, J.: From local to global patterns: Evaluation issues in rule learning algorithms. In: Morik, K., Boulicaut, J.F., Siebes, A. (eds.) Local Pattern Detection. pp. 20–38. Springer-Verlag (2005)

Fürnkranz, J., Gamberger, D., Lavrač, N.: Foundations of Rule Learning. Springer-Verlag (2012)

Fürnkranz, J., Kliegr, T., Paulheim, H.: On cognitive preferences and the interpretability of rule-based models. arXiv preprint arXiv:1803.01316 (2018)

Gabriel, A., Paulheim, H., Janssen, F.: Learning semantically coherent rules. In: Cellier, P., Charnois, T., Hotho, A., Matwin, S., Moens, M.F., Toussaint, Y. (eds.) Proceedings of the 1st International Workshop on Interactions between Data Mining and Natural Language Processing co-located with The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2014). vol. 1202, pp. 49–63. CEUR Workshop Proceedings, Nancy, France (2014)

Gibaja, E., Ventura, S.: Multi-label learning: a review of the state of the art and ongoing research. Wiley Interdisciplinary Review: Data Mining and Knowledge Discovery 4(6), 411–444 (2014)

Gibaja, E., Ventura, S.: A tutorial on multilabel learning. ACM Comput. Surv. 47(3), 52 (2015)

Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: Advances in Knowledge Discovery and Data Mining (PAKDD 2004). pp. 22–30 (2004)

Goethals, B.: Frequent set mining. In: Maimon, O., Rokach, L. (eds.) The Data Mining and Knowledge Discovery Handbook, pp. 377–397. Springer-Verlag (2005)

Guo, Y., Gu, S.: Multi-label classification using conditional dependency networks. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Two. pp. 1300–1305. IJCAI'11, AAAI Press (2011)

Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining and Knowledge Discovery 8(1), 53–87 (2004)

Hayes, P.J., Weinstein, S.P.: CONSTRUE/TIS: A system for content-based indexing of a database of news stories. In: Rappaport, A.T., Smith, R.G. (eds.) Proceedings of the 2nd Conference on Innovative Applications of Artificial Intelligence (IAAI-90), May 1-3, 1990, Washington, DC, USA. pp. 49–64. IAAI '90, AAAI Press, Chicago, IL, USA (1991)

Herrera, F., Charte, F., Rivera, A.J., del Jesús, M.J.: Multilabel Classification - Problem Analysis, Metrics and Techniques. Springer (2016)

Hipp, J., Güntzer, U., Nakhaeizadeh, G.: Algorithms for association rule mining – a general survey and comparison. SIGKDD explorations 2(1), 58–64 (2000)

Janssen, F., Fürnkranz, J.: On the quest for optimal rule learning heuristics. Machine Learning 78(3), 343–379 (2010)

Knobbe, A.J., Crémilleux, B., Fürnkranz, J., Scholz, M.: From local patterns to global models: The LeGo approach to data mining. In: Knobbe, A.J. (ed.) From Local Patterns to Global Models: Proceedings of the ECML/PKDD-08 Workshop (LeGo-08). pp. 1–16. Antwerp, Belgium (2008)

Kralj Novak, P., Lavrač, N., Webb, G.I.: Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. Journal of Machine Learning Research 10, 377–403 (2009)

Lewis, D.D.: 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 Devlopment in Information Retrieval. pp. 37–50 (1992)

Lewis, D.D.: Reuters-21578 text categorization test collection distribution 1.0. README file (V 1.3) (2004)

Li, B., Li, H., Wu, M., Li, P.: Multi-label Classification based on Association Rules with Application to Scene Classification. In: Proceedings of the 2008 The 9th International Conference for Young Computer Scientists. pp. 36–41. IEEE Computer Society (2008)

Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Agrawal, R., Stolorz, P., Piatetsky-Shapiro, G. (eds.) Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98). pp. 80–86 (1998)

Liu, B., Ma, Y., Wong, C.K.: Improving an exhaustive search based rule learner. In: Zighed, D.A., Komorowski, H.J., Zytkow, J.M. (eds.) Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000). pp. 504–509. Lyon, France (2000)

Loza Mencía, E., Janssen, F.: Stacking label features for learning multilabel rules. In: Discovery Science - 17th International Conference, DS 2014, Bled, Slovenia, October 8-10, 2014, Proceedings, Lecture Notes in Computer Science, vol. 8777, pp. 192–203. Springer (2014)

Loza Mencía, E., Janssen, F.: Learning rules for multi-label classification: a stacking and a separate-and-conquer approach. Machine Learning 105(1), 77–126 (2016)

Malerba, D.: Learning recursive theories in the normal ilp setting. Fundamenta Informaticae 57(1), 39–77 (2003)

Malerba, D., Semeraro, G., Esposito, F.: A multistrategy approach to learning multiple dependent concepts. In: Machine Learning and Statistics: The Interface, chap. 4, pp. 87–106 (1997)

Minnaert, B., Martens, D., Backer, M.D., Baesens, B.: To tune or not to tune: Rule evaluation for metaheuristic-based sequential covering algorithms. Data Mining and Knowledge Discovery 29(1), 237–272 (2015)

Montañés, E., Senge, R., Barranquero, J., Quevedo, J.R., del Coz, J.J., Hüllermeier, E.: Dependent binary relevance models for multi-label classification. Pattern Recognition 47(3), 1494 – 1508 (2014)

Nemenyi, P.: Distribution-free multiple comparisons. Ph.D. thesis, Princeton University (1963)

Papagiannopoulou, C., Tsoumakas, G., Tsamardinos, I.: Discovering and exploiting deterministic label relationships in multi-label learning. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 915–924. KDD '15, ACM, New York, NY, USA (2015)

Park, S.H., Fürnkranz, J.: Multi-label classification with label constraints. In: Hüllermeier, E., Fürnkranz, J. (eds.) Proceedings of the ECML PKDD 2008 Workshop on Preference Learning (PL-08, Antwerp, Belgium). pp. 157–171 (2008)

Rapp, M.: A Separate-and-Conquer Algorithm for Learning Multi-Label Head Rules. Master thesis, TU Darmstadt, Knowledge Engineering Group (2016)

Rapp, M., Loza Mencía, E., Fürnkranz, J.: Exploiting anti-monotonicity of multi-label evaluation measures for inducing multi-label rules. In: Proceedings of the 22nd Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-18). Springer-Verlag (2018), to appear

Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine Learning 85(3), 333–359 (2011)

Senge, R., del Coz, J.J., Hüllermeier, E.: On the problem of error propagation in classifier chains for multi-label classification. In: Spiliopoulou, M., Schmidt-Thieme, L., Janning, R. (eds.) Proceedings of the 36th Annual Conference of the Gesellschaft für Klassifikation (GfKl-12). pp. 163–170. Hildesheim, Germany (2012)

Stecher, J., Janssen, F., Fürnkranz, J.: Shorter rules are better, aren't they? In: Calders, T., Ceci, M., Malerba, D. (eds.) Proceedings of the 19th International Conference on Discovery Science (DS-16). pp. 279–294. Springer-Verlag (2016)

Sucar, L.E., Bielza, C., Morales, E.F., Hernandez-Leal, P., Zaragoza, J.H., Larrañaga, P.: Multi-label classification with Bayesian network-based chain classifiers. Pattern Recognition Letters 41, 14–22 (2014)

Sulzmann, J.N., Fürnkranz, J.: A comparison of techniques for selecting and combining class association rules. In: Knobbe, A.J. (ed.) From Local Patterns to Global Models: Proceedings of the ECML/PKDD-08 Workshop (LeGo-08). pp. 154–168. Antwerp, Belgium (2008)

Thabtah, F., Cowling, P., Peng, Y.: MMAC: A New Multi-Class, Multi-Label Associative Classification Approach. In: Proceedings of the 4th IEEE ICDM. pp. 217–224 (2004)

Thabtah, F., Cowling, P., Peng, Y.: Multiple labels associative classification. Knowledge and Information Systems 9(1), 109–129 (2006)

Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.P.: Multilabel classification of music into emotions. In: Proc. 9th International Conference on Music Information Retrieval (ISMIR 2008). pp. 325–330 (2008)

Tsoumakas, G., Katakis, I., Vlahavas, I.P.: Mining Multi-label Data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 667–685 (2010)

Tsoumakas, G., Zhang, M., Zhou, Z.: Introduction to the special issue on learning from multi-label data. Machine Learning 88(1-2), 1–4 (2012)

Varma, M., Cissé, M. (eds.): Proceedings of the NIPS-15 Workshop on Extreme Classification: Multi-class and Multi-label Learning in Extremely Large Label Spaces (XC-15) (2015)

Veloso, A., Meira, Jr., W., Gonçalves, M., Zaki, M.: Multi-label lazy associative classification. In: Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases. pp. 605–612. PKDD 2007 (2007)

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

Webb, G.I.: Recent progress in learning decision lists by prepending inferred rules. In: Proceedings of the 2nd Singapore International Conference on Intelligent Systems. pp. B280–B285 (1994)

Webb, G.I.: Efficient search for association rules. In: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2000). pp. 99–107. Boston, MA (2000)

Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of association rules. In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97). pp. 283–286. Newport, CA (1997)

Zhang, C., Zhang, S.: Association Rule Mining: Models and Algorithms. Springer-Verlag (2002)

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