Stacking Label Features for Learning Multilabel Rules*

Eneldo Loza Mencía and Frederik Janssen

Technische Universität Darmstadt
Knowledge Engineering Group
{eneldo,janssen}@ke.tu-darmstadt.de

Abstract. Dependencies between the labels is commonly regarded as the crucial issue in multilabel classification. Rules provide a natural way for symbolically describing such relationships, for instance, rules with label tests in the body allow for representing directed dependencies like implications, subsumptions, or exclusions. Moreover, rules naturally allow to jointly capture both local and global label dependencies.
We present a bootstrapped stacking approach which uses a common rule learner in order to induce label-dependent rules. For this, we learn for each label a separate ruleset, but we include the remaining labels as additional attributes in the training instances. Proceeding this way, label dependencies can be made explicit in the rules. Our experiments show competitive results in terms of the standard multilabel evaluation measures. But more importantly, using these additional attributes is shown to allow to discover and consider label relations as well as to better comprehend the available multilabel datasets.
However, this approach is only a first step towards integrating the multilabel rule learning directly in the rule induction process, e.g., in typical separate-and-conquer rule learners. We present future perspectives, advantages, and arising issues in this regard.

1 Introduction

Rule learning has a very long history and is a well-known problem in the machine learning community. Over the years many different algorithms to learn a set of rules were introduced. The main advantage of rule-based classifiers are interpretable models as rules can be easily comprehended by humans. Also, 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 multi-class classification.

On the other hand, many problems involve assigning more than a single class to an object. These so-called multilabel problems can often be found when text is classified

* This is the authors' version of the work retrieved from http://www.ke.tu-darmstadt.de. The original publication is available at http://www.springerlink.com, DOI: 10.1007/978-3-319-11812-3_17, and appeared in Džeroski, Sašo et al. (Eds.): *Discovery Science 17th International Conference, DS 2014, Bled, Slovenia, October 8-10, 2014. Proceedings*, LNAI 8777, pp. 192–203, 2014.

into topics or tagged with keywords, but there are also many examples from other media such as the recognition of music instruments or emotions in audio recordings or the classification of scenes in images.

It is widely accepted that one major issue in learning from multilabel data is the exploitation of label dependencies. Learning algorithms may greatly benefit from considering label correlations, and we believe that rule induction algorithms provide a good base for this. Firstly, so called global dependencies between only labels can be explicitly modeled and expressed in form of rules. But also, and much more interesting, dependencies that include both label and regular features can be constituted, which we refer to as local dependencies. Secondly, such rules are directly interpretable and comprehensible for humans. Even if complex and long rules are generated, the implication between classes can be estimated more easily than with other approaches by focusing on the part of the rules that considers the classes. Hence, one is able to directly analyze the induced rule models and may greatly benefit from these explicit notations, in contrast to other types of models where the key information is not accessible directly.

We propose in this work to learn such interdependencies by providing the true label information directly to the rule learner. This is done by stacking the label features as additional input instance features. Although this is not the first work in making use of stacking in order to consider label dependencies (cf. Sec. 3.1), it is to our knowledge the first time that rule induction was used in order to make the label dependencies explicit. We show that the proposed method, though conceptually very simple, is suitable in order to reveal global as well as local label dependencies. Almost more importantly, the induced models allow for a detailed analysis of the datasets commonly used in the community for benchmarking w.r.t. the contained dependencies.

The proposed bootstrapping in the prediction phase remains open for discussion, though its performance is competitive to straight-forward approaches. But our ultimate goal is to have a complete framework for multilabel rule induction instead of employing special schemes for learning and predicting. We give some perspectives and ideas for further research in the end of the paper.

2 Multilabel Classification and Inductive Rule Learning

2.1 Multilabel Classification

Multilabel classification refers to the task of learning a function $h(\mathbf{x})$ that maps instances $\mathbf{x} = (x_1, \ldots, x_m) \in \mathscr{X}$ to label subsets or label vectors $\mathbf{y} = (y_1, \ldots, y_n) \in \{0,1\}^n$, where $\mathscr{L} = \{\lambda_1, \ldots, \lambda_n\}$, $n = |\mathscr{L}|$ is a finite set of predefined labels and where each label attribute y_i corresponds to the absence (0) or presence (1) of label λ_i . Thus, in contrast to multiclass learning, alternatives are not assumed to be mutually exclusive, such that multiple labels may be associated with a single instance. This, and especially the resulting correlations and dependencies between the labels in $\mathscr{L}$, make the multilabel setting particularly challenging and interesting compared to the classical field of binary and multiclass classification.

From a probabilistic point of view, this is one of the main differences. In binary and multiclass problems the only observable probabilistic dependence is between the input

variables, i.e., the attributes x_j, and the label variables y_i. A learning algorithm tries to learn exactly this dependence in form of a classifier function h. In fact, if a classifier provides a score or confidence for its prediction $\hat{\mathbf{y}} = h(\mathbf{x})$, this is often regarded as an approximation of $P(\mathbf{y} = \hat{\mathbf{y}} \mid \mathbf{x})$, i.e., the probability that $\hat{\mathbf{y}}$ is true given a document $\mathbf{x}$.

From the early beginning of multilabel classification, there have been attempts to exploit these types of *label correlations* [e.g. 12, 7, 17]. A middle way is followed by Read et al. [14] and Dembczyński et al. [5] and their popular (probabilistic) classifier chains by stacking the underlying binary relevance classifiers with the predictions of the previous ones. However, only recently Dembczyński et al. provided a clarification and formalization of label dependence in multilabel classifications. Following their argumentation, one must distinguish between unconditional and conditional label dependence. Roughly speaking, the *unconditional dependence* or independence between labels does not depend on a specific given input instance (the condition) while *conditional dependence* does. We may also refer to these as *global* and *local* dependencies, since they are revealed globally or only in subspaces of the input space.

An example may illustrate this: Suppose a label space indicating topics from news articles and a subtopic *foreign affairs* of the topic *politics*. Obviously, there will be a dependency between both labels, since the presence of a subtopic implies the presence of the super topic and the probability of *foreign affairs* would be higher than average if *politics* is observed. These probabilities are *unconditional* or *global* since they do not depend on a particular document. Suppose now that a particular news article is about the *Euro crisis*. Under this condition, the *conditional* probabilities for both labels as well as the dependency between would likely increase and hence be different from the unconditional ones. However, if an article was about the *cardiovascular problems of Ötzi*, we would observe that both labels are *conditionally independent* for this instance, since the probability for one label would very likely not depend on the presence of the other label (both being very low).

The predominant approach in multilabel classification is *binary relevance* (BR) learning [cf. e.g. 16]. It tackles a multilabel problem by learning one classifier for each label, using all objects of this label as positive examples and all other objects as negative examples. There exists hence a strong connection to concept learning, which is dedicated to infer a model or description of a target concept from specific examples of it [see, e.g., 4]. When several target concepts are possible or given for the same set of instances, we formally have a multilabel problem. The problem of this approach is that each label is considered independently of each other, and as we have seen by the example given before, this can lead to loss of useful information for classification.

A possible simple solution to generate rules that may consider several labels in the head is to use the label powerset (LP) transformation [cf. 16], which decomposes the initial problem into a multiclass problem with $\{\mathbf{y} \mid (\mathbf{x},\mathbf{y}) \in \text{training set}\} \subseteq \{0,1\}^n$ as possible classes. This problem can then be processed with common rule induction algorithms, which will thus produce rules with several labels in the head.

This approach is potentially able to consider conditional dependencies, namely the case of label co-occurrences. The main drawback is that the number of classifiers that have to be learned grows exponentially. Another obvious disadvantage is that we can only predict label relations and combinations which were seen in the training data.

2.2 Inductive Rule Learning

As the goal of this paper is to make label dependencies explicit by using rules, we will also shorty introduce inductive rule learning. This is one of the oldest and therefore best researched fields in machine learning. Many algorithms were proposed over the years, *Ripper* [3] being one of the most popular and used ones. In this work, we used this algorithm, but the proposed method does also naturally work with other rule learning algorithms. *Ripper* is a so-called separate-and-conquer (SeCo) algorithm [6], i.e., it proceeds by learning a good rule on the data, then adds the rule to the ruleset, removes all examples covered by this rule, and searches the next one as long as (positive) examples are left in the dataset. 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. SeCo usually only works for binary datasets. Hence, a natural way of addressing multilabel problems is to consider each label separately (cf. BR), resulting in a model consisting of separate rulesets for each label.

2.3 Different Forms of Multilabel Rules

A rule learner has a set of rules (*ruleset*) as result. These rules are of the form

$$head \leftarrow body$$

where the body consists of a number of *conditions* (attribute-value tests) and, in the regular case, the head has only one single condition of the form $y_i = 0$ or 1 (in our case). We refer to this type of rules as *single-label head* rules in contrast to *multi-label head* rules, which contain several label assignments in their head and can thus conveniently express label co-occurrences. Commonly, the conditions in the body are on attributes from the instance space. However, in order to reflect label dependencies (e.g., implications, subsumptions, or exclusions), we would need to have labels on both sides of the rule. Hence, if a rule may contain conditions on the labels, we refer to it as label-dependent rules (also referred to as *contextual* rules [4]), and *label-independent* if this is not the case. Global dependencies are hence best reflected by *full label-dependent* bodies, whereas local dependencies can be described by *partially label-dependent* rules with mixed attributes in the body.

In summary: We start from **label-independent single-label** rules. Label dependencies can already be captured by **label-independent multi-label** rules. The next section describes a straight-forward approach for obtaining such rules. Future extensions are proposed in Sec. 6. This particular work focuses on learning **label-dependent single-label** rules (Sec. 3), which, as shown, are well suited for modeling and expressing label dependencies. The full expressiveness is though obtained by **label-dependent multi-label** rules, which we leave for further research (Sec. 6).

3 Learning Label-Dependent Rules

We present in the next subsection a straight-forward, yet effective approach in order to learn label-dependent rules which allows to discover valuable information in data.

3.1 Stacking of Label Features

The recently very popular classifier chains [14] were found to be an effective approach for exploiting conditional label dependencies. Classifier chains (CC) make use of stacking the previous BR classifiers' predictions in order to implement the chain rule $P(y_1,\ldots,y_n) = P(y_n \mid y_1,\ldots,y_{n-1})$ in probability theory, since they learn the binary classifiers h_i with training examples of the form $(x_1,\ldots,y_1,\ldots,y_{i-1})$ [cf. 5]. One drawback of CC is the (randomly chosen) predetermined, fixed order of the classifiers (and hence the labels) in the chain, which makes it impossible to learn dependencies in the contrary direction. This was already recognized by D. Malerba and Esposito [4] in 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.

Thus, we propose to use a full stacking approach in order to overcome the main disadvantage of CC, i.e., the fixed order. Like in binary relevance, we learn one theory for each label, but we expand our training instances by the label information of the other labels, i.e., the training examples vectors for learning label y_i are of the type $(x_1,\ldots,y_1,\ldots,y_{i-1},y_{i+1},\ldots,y_n)$ for an instance $\mathbf{x}$. The result of using this as training data is exactly what we are seeking for, namely label-dependent single-label rules. The amount of label-features in the body additionally allows us to determine the type of dependency. We refer to this technique as *stacked binary relevance* (SBR) in contrast to plain, unstacked BR.

This is very similar to the approaches of Godbole and Sarawagi [8], Guo and Gu [9], and very recently, Montañés et al. [13]. They all have in common that they are using label presence information (either directly from the training data, or from the outputs of underlying BR classifiers) as (either sole or additional) features in order to learn an ensemble of binary relevance classifiers on top. The closest related approaches to our proposition are the conditionally dependency networks (CDN) [9] and the dependent binary relevance (DBR) models [13]. Both learn their models as indicated before but with one major difference: Since they are 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. To make the dependencies explicit and at the same time keep a high prediction quality, we propose to use rule-based models. One additional difference between the approaches is how the prediction is conducted, which is discussed next.

3.2 Prediction by Bootstrapping

For the prediction we propose to use a bootstrapping approach in the sense that we apply our models iteratively on our own previous predictions until the predictions are stable or any other stopping criterium is met. More formally, we use the learned models

h'_i to produce a prediction $\hat{\mathbf{y}}_j = (\hat{y}_{j,1}, \hat{y}_{j,2}, \ldots)$ where $\hat{y}_{j,i} = h'_i(\mathbf{x}, \hat{\mathbf{y}}_{j-1})$ is based on the predictions in the previous iteration $j-1$.

One obvious issue with this approach is the initialization of $\hat{y}_0$. A possible option, also proposed by DBR, is to use the predictions of a BR ensemble, i.e., $\hat{y}_{0,i} = h_i(\mathbf{x})$. We also evaluate the option of initializing with *unknown* label information, i.e., $\hat{\mathbf{y}}_0 = (?, ?, \ldots)$, and to benefit from the natural support of symbolic approaches for such attribute states (*missing*, *don't care*, etc.). On the other hand, this approach only works if the rule learner found enough rulesets with label-independent rules so that the bootstrapping can proceed, which is in fact somehow contradictory to the objective of detecting as much dependencies as possible. In the future, we also plan to use random initialization. Together with enough iterations of *Gibbs sampling*, this was shown to be very effective for CDN.

We also may make use of an additional capability of rule learners, namely to abstain from classifying if no appropriate rule was found (instead of predicting the default rule) so that the label attribute may be filled up in consequent iterations.

4 Evaluation

An overview of the used datasets[1] is given in Tab. 1. They are from different domains and have varying properties. Details of the data are given in the analysis when needed. As rule learner, we use the JRip implementation of Ripper [3] with default parameters, except for the pruning, which is turned off or on depending on the experiment.

We use micro-averaged precision and recall to evaluate our results, i.e., we compute a two-class confusion matrix for each label ($y_i = 1$ vs. $y_i = 0$) and eventually aggregate the results by (component-wise) summing up all n matrices into one global confusion matrix (cf. [16]). Recall and precision is computed based on this global matrix in the usual way, F1 denotes the unweighted harmonic mean between precision and recall. In addition, we measure the subset accuracy, which is the percentage $\frac{1}{m}\sum_{i=1}^{m}[[\mathbf{y}_i = \hat{\mathbf{y}}_i]]$ of the m test instances for which the labelsets were exactly correctly predicted ($[[z]]$ returns 1 if z is true, otherwise 0). The measures, as well as other statistics, are averaged over the ten-fold cross validation results, which we use for all our experiments.

4.1 Model and Data Analysis

Tab. 2 shows the properties of the rulesets generated by using plain BR and stacked BR decomposition with JRip. As we will see in the following, these statistics not only help to analyze the algorithm, but even more importantly, they are of great use for analyzing and understanding the datasets at hand. Though it is commonly assumed that there exist label dependencies between the labels in multilabel datasets, and many works deal with exploiting such dependencies, this assumption is most often not explicitly examined. To our knowledge, this is the first work providing a systematic analysis of the label dependencies contained in seven of the most popular benchmarks.

[1] We refer to the MULAN repository for details and sources: http://mulan.sf.net/datasets.html.

Table 1. 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 labelsets in the data.

name	domain	instances	nominal	numeric	labels	cardinality	density	distinct
EMOTIONS	music	593	0	72	6	1.87	0.311	27
SCENE	image	2407	0	294	6	1.07	0.179	15
YEAST	biology	2417	0	103	14	4.24	0.303	198
GENBASE	biology	662	1186	0	27	1.25	0.046	32
MEDICAL	text	978	1449	0	45	1.25	0.028	94
ENRON	text	1702	1001	0	53	3.38	0.064	753
CAL500	music	502	0	68	174	26.0	0.150	502

Column (5) shows the percentage of conditions on labels w.r.t. to all conditions in the model. We see that there is a great divergence between the datasets. E.g., the models for GENBASE do not use label features at all, i.e., their rules' bodies are completely label-independent. This is a strong indicator that we have completely independent labels in this dataset, or, at least, very weak dependencies. This is remarkable, since this breaks the main assumption, mentioned before, and yet this dataset may have often been used in the literature to show the ability of a certain algorithm to exploit label dependencies. In this case though, learning each label independently is already sufficient and exploiting (possibly non-existing) label dependencies clearly will not yield better performance. A look into columns (1)-(4), the prediction quality (Tab. 3) and eventually into the models, reveals that the presence of one single short amino acid chain (instance feature) is often enough to correctly predict a particular functional family (label).

For (5) it is also remarkable that pruning substantially increases the percentage of used label features. Pruning tries to remove conditions and rules which work good on a training set, but do not generalize well on a separate validation set. Hence, this increase indicates that label features are more useful for obtaining more general models than the original instance features. However, the increase does not come hand in hand with a decrease in the size of the models comparing BR and stacked BR, as can be seen by the average size of the rulesets (columns (1) and (2)) and rules ((3) and (4)), which does not reveal any trend.

While (5) may serve as an indicator of general dependency between labels, columns (6) and (7) allow to further differentiate. E.g., 20.8% of fully label-dependent rulesets for YEAST, i.e., rulesets with rules only having conditions on label features, show that (at least) 20.8% of the labels in YEAST are *unconditionally* dependent on other labels. On the other hand, by leaving out the 6.2% of labels which are independent, we can derive that (at most) 73.0% of the labels are *conditionally* dependent on other labels. Note that (6) should be considered as a lower bound, since the rate substantially suffers from the high number of instance features due to a kind of *instance feature flooding*: The probability of selecting an instance feature in the refinement step of a rule instead of an equally good label feature increases with growing number of instance features. However, the same effect cannot be observed for (7).

Table 2. Statistics. From left to right, for BR model: (1) avg. # rules per label ruleset, (2) avg. # conditions per rule. For stacked model: (3) avg. # rules per label ruleset, (4) avg. # conditions per rule, (5) percentage of conditions with label feature tests, (6) perc. of label rulesets depending *only* on other labels, (7) perc. of label rulesets depending *only* on instance features.

dataset	pruning	(1)	(2)	(3)	(4)	(5)	(6)	(7)
EMOTIONS	yes	3.26	2.78	2.74	3.09	35.0%	18.0%	0.0%
EMOTIONS	no	11.50	4.02	11.02	4.18	17.6%	0.0%	0.0%
SCENE	yes	6.72	4.27	5.44	4.44	16.0%	0.0%	18.0%
SCENE	no	13.58	5.40	11.10	5.09	10.2%	0.0%	2.0%
YEAST	yes	2.47	3.72	3.78	2.56	63.0%	20.8%	6.2%
YEAST	no	7.20	5.95	10.58	3.78	31.3%	0.0%	0.0%
GENBASE	yes	0.90	1.05	0.90	1.05	0.0%	0.0%	100.0%
GENBASE	no	0.99	1.29	0.99	1.29	0.0%	0.0%	100.0%
MEDICAL	yes	1.08	1.72	1.07	1.81	17.4%	0.0%	79.3%
MEDICAL	no	2.46	3.47	2.00	3.17	13.6%	0.0%	73.6%
ENRON	yes	1.54	3.38	1.89	3.37	35.9%	3.3%	35.0%
ENRON	no	5.82	4.97	6.94	4.68	25.1%	0.0%	11.0%
CAL500	yes	0.45	2.23	1.37	2.07	60.7%	29.0%	23.8%
CAL500	no	6.03	3.88	6.82	3.51	29.7%	1.2%	1.7%

The datasets with the highest observed degree of label dependency are YEAST and CAL500. For CAL500, this may be explained by the categorizations of songs into emotions, which often come hand in hand or completely contradict, like *Angry-Agressive* against *Carefree-Lighthearted*.

Examples of learned rulesets for YEAST are given in Fig. 1. In this particular case, we see a much more compact and less complex ruleset for *Class4* for the stacked model than for the independently learned BR classifier. The ruleset also seems more appropriate for a domain expert to understand coherences between proteins (instance features) and protein functions (labels).

Fig. 1 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 stacked model reads very well, and the

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

Fig. 1. Example rulesets for one exemplary label, respectively, learned by the normal and the stacked BR approach. Attribute names in italic denote label attributes, attributes with an overline denote negated conditions.

found relationship seems to be even comprehensible by non-experts: 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*.

In ENRON, which is concerned with the categorization of emails during the Enron scandal, the model is less comprehensible, as it is also for the BR model. However, the relation between *Personal* and *Joke* can clearly be explained from the hierarchical structure on the topics. This also shows the potential of using rule learning in multilabel classification for reconstructing underlying hierarchies.

4.2 Prediction Performance

Tab. 3 shows the predictive performance of the different approaches. We compare BR, LP, Stacked BR with BR initialization and abstaining ($\text{SBR}_{BR/?}$) or predicting the default label ($\text{SBR}_{BR/d}$), respectively, in the case of the default rule firing, and lastly SBR with empty initialization and abstaining ($\text{SBR}_{?/?}$). For all approaches, we used the pruning version of JRip. Due to the space limit, we only report the results after the 10^{th} bootstrapping iteration in the case of SBR.[2]

As expected, LP is the best approach w.r.t. subset accuracy. Somehow surprisingly, BR and both first SBRs obtain very similar avg. ranks, although the stacking of the label features is considered to particularly address the correct prediction of labelsets [5, 9, 13]. $\text{SBR}_{?/?}$ clearly suffers from the cold start problem when many label dependencies were encountered, best seen by the high precision but very low recall and subset accuracy obtained. BR is best for precision, but is always worse than $\text{SBR}_{BR/?}$ and $\text{SBR}_{BR/d}$ on recall,[3] which in general find the better trade-off between recall and precision, beating all other approaches on F1. Recall that BR's predictions are inputs for $\text{SBR}_{BR/?}$ and $\text{SBR}_{BR/d}$. Apparently, the additional iterations applying the stacked models allow labels which were initially missed to be found due to the label context.

5 Related Work

Many rule-based approaches to multilabel learning rely on association rules as those can have many conditions in the head. However, as the goal is classification, usually Classification Association Rules (CARs) are used, instead of regular association rules that would also find relations between instance-features. E.g., in Ávila et al. [2] a genetic algorithm is used to induce single-label association rules. A multilabel prediction is then built by using a combination of all covering rules of the BR rule sets. A good distribution of the labels is also ensures by using a token-based re-calculation of the fitness value of each rule. Li et al. [10] learn single-label association rules as well. For prediction, exactly those labels are set that have a probability greater than 0.5 in the covering rules.

[2] We found that more iterations consistently decrease subset acc. and recall, but increases precision and F1. However, the average absolute difference was consistently below 1%.

[3] Except of course for GENBASE, where all plain and stacked BR models are equal.

Table 3. Experimental performance on the seven datasets. The small number after the result indicates the rank of the particular approach. The last block shows the average over these ranks.

Approach	Subset Acc.	Precision	Recall	F1	Subset Acc.	Precision	Recall	F1
	SCENE				EMOTIONS			
BR	46.24% 2	68.82% 2	60.94% 4	64.55% 3	23.60% 3	65.54% 2	57.23% 3	60.97% 3
LP	58.33% 1	63.61% 4	61.09% 3	62.32% 4	20.56% 4	56.47% 5	55.66% 4	56.01% 4
$SBR_{BR/?}$	46.11% 3	65.56% 3	65.68% 2	65.58% 2	24.28% 2	64.02% 3	62.73% 2	63.24% 2
$SBR_{BR/d}$	45.32% 4	58.31% 5	77.79% 1	66.63% 1	24.96% 1	57.46% 4	75.54% 1	65.24% 1
$SBR_{?/?}$	29.13% 5	75.83% 1	33.28% 5	46.08% 5	9.09% 5	70.03% 1	21.97% 5	32.54% 5
	GENBASE				MEDICAL			
BR	96.83% 2.5	98.95% 2.5	98.42% 2.5	98.68% 2.5	66.96% 2	80.26% 2	84.29% 3	82.19% 1
LP	95.77% 5	97.30% 5	94.78% 5	95.99% 5	68.20% 1	80.18% 3	73.97% 4	76.93% 4
$SBR_{BR/?}$	96.83% 2.5	98.95% 2.5	98.42% 2.5	98.68% 2.5	66.86% 3	79.38% 4	84.78% 2	81.96% 2
$SBR_{BR/d}$	96.83% 2.5	98.95% 2.5	98.42% 2.5	98.68% 2.5	66.25% 4	78.21% 5	86.01% 1	81.89% 3
$SBR_{?/?}$	96.83% 2.5	98.95% 2.5	98.42% 2.5	98.68% 2.5	28.93% 5	82.39% 1	36.16% 5	50.13% 5
	ENRON				CAL500			
BR	9.17% 3	62.75% 1	49.09% 3	55.03% 2	0.00% 3	52.73% 1	24.88% 4	33.76% 3
LP	11.51% 1	41.06% 5	15.11% 4	22.08% 4	0.00% 3	31.90% 4	31.80% 1	31.84% 4
$SBR_{BR/?}$	9.17% 4	57.96% 2	55.09% 2	56.40% 1	0.00% 3	47.61% 2	30.90% 2	37.42% 1
$SBR_{BR/d}$	9.87% 2	43.13% 4	59.06% 1	49.71% 3	0.00% 3	44.76% 3	30.43% 3	36.20% 2
$SBR_{?/?}$	0.06% 5	53.10% 3	7.50% 5	13.08% 5	0.00% 3	26.48% 5	0.26% 5	0.51% 5
	YEAST				Average rank			
BR	9.18% 4	68.47% 1	55.33% 4	61.19% 3	2.79 3	1.64 1	3.36 3	2.50 3
LP	16.92% 1	60.04% 4	57.10% 3	58.52% 4	2.29 1	4.29 5	3.43 4	4.14 4
$SBR_{BR/?}$	10.18% 2.5	66.88% 2	57.63% 2	61.90% 2	2.86 4	2.64 3	2.07 2	1.79 1
$SBR_{BR/d}$	10.18% 2.5	58.31% 5	66.21% 1	61.98% 1	2.71 2	4.07 4	1.50 1	1.93 2
$SBR_{?/?}$	0.25% 5	65.35% 3	1.31% 5	2.56% 5	4.36 5	2.36 2	4.64 5	4.64 5

A different idea is to introduce multi-label instead of single-label rules. Those are able to directly classify a multi-label-instance without the need to combine single-label rules [1]. Interestingly, the proposed rules also allow for postponing the classification by offering a "*don't care*"-value. The classification is then done by using a weighted voting scheme as many multilabel rules may cover the example.

Another multilabel rule algorithm is *MMAC* [15]. Here a multi-class, multilabel associative classification approach is used by not only generating from all frequent itemsets the rules that pass the confidence threshold but also include the second best rules and so on. Multilabel rules are then generated from these association rules by the frequent itemsets where covered instances are removed then. Rules with same conditions are then merged which enables a total ranking of all labels for each test instance.

Other approaches are from the inductive logic programming field. Here, some also allow for having label features in the rule bodies, but due to the different nature disclosed by relational rules, these methods are not in the scope of this paper. In summary, label dependencies are not tackled explicitly though they might be taken into account by algorithm-specific properties. Please consider [11] for a more extensive discussion.

6 Future Challenges

All presented approaches for learning multilabel models, BR, LP and SBR decomposition, have one aspect in common, namely that they transform the original problem into several subproblems, which are then solved independently. This might be appropriate or even advantageous for certain use cases, for instance when the objective is to obtain isolated theories representing each label (cf. concept learning), or w.r.t. efficiency. But often it is more desirable to obtain one global theory comprehensively explaining a particular multilabel dataset. The induction of one global model also allows a better control over the objective loss, an important issue in multilabel classification due to the variety of existing measures, resulting directly from the diversity of the real life scenarios.

Regarding the introduced stacked BR approach which we used for learning label-dependent single-label rules, we propose to integrate the stacking of label features directly into the SeCo induction process. The idea is to start with unset label features, consequently only label-independent rules will be learnt in the beginning. However, the covered rules are not separated, but labeled accordingly and readded to the training set. Hence, we would get rid of the cold start and deadlock problem and no bootstrapping or sampling would be necessary.

Multiple labels in the head allow for representing co-occurrence relationships. In addition, only label-dependent multi-label rules allow to express all types of possible dependencies. The solution using LP can learn multilabel head rules, but with the mentioned shortcomings (Sec. 2.1). Therefore, we propose the following modifications.

In order to obtain one single global theory, we learn so-called multiclass decision lists, which allow to use different heads in consecutive rules of the decision list. If we limit ourselves to labelsets seen in the training data, this corresponds to using LP transformation with a multiclass decision list learner. However, the evaluation for each possible labelset can be very expensive ($\mathcal{O}(2^n)$ in the worst case). The following greedy approach may solve this. It starts by evaluating the condition candidates w.r.t. to each label independently in order to determine the best covered label. Having selected the best covered label for the given rule body, we can only stay the same or get worse if we now add an additional label to our head, since the number of covered examples remain the same and the number of covered *positives*, for which the head applies, cannot increase. Hence, depending on the heuristic used, we can safely prune great part of the label combinations by exploiting the anti-monotonicity of the heuristic.

Challenges to both proposed extensions, and to the self-evident combination of both, concern the rule learning process itself: The right selection of the heuristic was already a complex issue in traditional rule induction and has to be reviewed for multilabel learning. Furthermore, using unordered and multiclass decision lists gain new relevance, too.

We plan to use our method in order to analyze the datasets, and further benchmark datasets commonly used in the literature, in more detail. Regarding prediction quality, we expect to improve our performance by adopting the extensions presented in Sec. 3. An extended empirical study with additional state-of-the-art algorithms would reveal any development and allow further comparisons.

7 Conclusions

In this work, we introduced a simple yet effective approach to making label dependencies explicit with the means of rules. The proposed stacking approach is able to induce rules with labels as conditions in the bodies of the rules. In our analyses on seven multilabel datasets, the resulting models turned out to be indeed very useful in order to discover interesting aspects a normal rule learner is unable to uncover. For instance, we found out that the GENBASE dataset exhibits only very weak label dependencies, if any at all, despite the fact that it is frequently used for evaluating multilabel algorithms. In contrast to other approaches, the proposed method naturally allows for discovering and expressing local as well as global label dependencies.

The second part of the evaluation showed that our approach works particularly well for trading-off recall and precision, obtaining the best result w.r.t. F-measure. For subset accuracy, it is beaten by LP, which is particularly tailored towards this measure. However, the introduced technique of bootstrapping predictions still requires the initial input of a plain BR. Therefore, we presented two different but combinable directions for learning global theories as future challenges in the field of multilabel rule learning.

References

[1] 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)

[2] 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)

[3] Cohen, W.W.: Fast Effective Rule Induction. In: Proceedings of the 12th International Conference on Machine Learning (ICML-95). pp. 115–123 (1995)

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

[5] 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)

[6] Fürnkranz, J.: Separate-and-Conquer Rule Learning. Artificial Intelligence Review 13(1), 3–54 (February 1999)

[7] Ghamrawi, N., McCallum, A.: Collective multi-label classification. In: CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management. pp. 195–200. ACM (2005)

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

[9] 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)

[10] 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)

[11] Loza Mencía, E., Janssen, F.: Towards multilabel rule learning. In: Proceedings of the German Workshop on Lernen, Wissen, Adaptivität - LWA2013. pp. 155–158 (2013)

[12] McCallum, A.K.: Multi-label text classification with a mixture model trained by EM. In: AAAI 99 Workshop on Text Learning (1999)
[13] 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)
[14] Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine Learning 85(3), 333–359 (2011)
[15] 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)
[16] Tsoumakas, G., Katakis, I., Vlahavas, I.P.: Mining Multi-label Data. In: Data Mining and Knowledge Discovery Handbook, pp. 667–685 (2010)
[17] Zhu, S., Ji, X., Xu, W., Gong, Y.: Multi-labelled classification using maximum entropy method. In: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 274–281. ACM (2005)