

---

# Implementation and Hyperparameter Tuning of a Separate-and-Conquer Multi-Label Rule Learner

---

Bachelor thesis by Jakob Steeg  
Date of submission: 06.10.2020

1. Review: Dr. Eneldo Loza Mencía  
2. Review: Michael Rapp  
Darmstadt



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Computer Science  
Department  
Knowledge Engineering  
Group

---

## Abstract

---

Multi-label classification is a classification task assigning an arbitrary number of labels to an object. For example, multiple genres like *comedy* or *sci-fi* may be assigned to a single movie. This work proposes an implementation of an algorithm for learning multi-label rule lists. Multi-label rule lists are human understandable classification models, which describe the relations between data and the labels assigned to it. Even though many properties of single-label rule learning algorithms have been studied, not many works examine the optimal configurations for multi-label head rule learning algorithms. This work introduces an implementation of a multi-label head rule learner and presents experiments which determine the optimal hyperparameter configurations for different environments. The implemented algorithm is based on an algorithm proposed by Loza Mencía and Janssen (2016), adapted by Rapp (2016) to learn multi-label head rules and extended to further favor multi-label head rules using relaxed pruning by Klein (2018). The proposed implementation incorporates multiple approaches suggested in related work and suggests algorithmic improvements over a naive implementation of the base algorithm. The evaluation of the predictive performance of the implemented algorithm shows that the algorithm is capable of efficiently learning relatively short rule lists with a performance comparable to other multi-label classification algorithms.

---

# Contents

---

<b>1. Introduction</b>	<b>8</b>
1.1. Multi-Label Classification . . . . .	8
1.2. Goal of the Thesis . . . . .	8
1.3. Organization of the Thesis . . . . .	9
<b>2. Fundamentals</b>	<b>10</b>
2.1. Multi-Label Classification . . . . .	10
2.1.1. Problem Transformation . . . . .	11
2.1.2. Label Dependencies . . . . .	11
2.2. Inductive Rule Learning . . . . .	12
2.2.1. Rule Types . . . . .	12
2.2.2. Separate-and-Conquer Rule Learning . . . . .	12
2.2.3. Decision Lists . . . . .	13
2.2.4. Default Rule . . . . .	13
2.3. Evaluation Metrics . . . . .	14
2.3.1. Bipartition Evaluation . . . . .	14
2.3.2. Multi-Label Evaluation . . . . .	15
2.3.3. Selected Metrics . . . . .	16
2.3.4. Aggregation and Averaging . . . . .	17
2.4. Pruning the Label Space . . . . .	18
2.4.1. Anti-Monotonicity . . . . .	19
2.4.2. Decomposability . . . . .	19
2.4.3. Relaxed Pruning . . . . .	20
<b>3. Learning Multi-Label Rules</b>	<b>22</b>
3.1. Rule Induction . . . . .	23
3.2. Refining Rule Conditions . . . . .	26
3.2.1. Aggregating Confusion Matrices . . . . .	28
3.2.2. Sparse Condition Finding . . . . .	29
3.3. Head Refinement . . . . .	30
3.3.1. Single-Label Head Refinement . . . . .	31
3.3.2. Multi-Label Head Refinement . . . . .	31
3.4. Measuring Performance . . . . .	32
<b>4. Evaluation of Results</b>	<b>34</b>
4.1. Experimental Setup . . . . .	34
4.1.1. Bootstrap Bias Corrected Cross Validation . . . . .	34
4.1.2. Data Sets . . . . .	34



- 4.1.3. Tuned Parameters . . . . . 35
- 4.2. Predictive Quality . . . . . 36
  - 4.2.1. Single-Label Head Refinement . . . . . 37
  - 4.2.2. Multi-Label Head Refinement . . . . . 37
- 4.3. Model Characteristics . . . . . 39
  
- 5. Conclusion . . . . . 41**
  - 5.1. Summary . . . . . 41
  - 5.2. Future Work . . . . . 41
  
- Appendices . . . . . 42**
  - A. Performance Evaluations . . . . . 42**
  - B. Model Characteristics . . . . . 52**

---

## List of Figures

---

1.	Prunable search for the best multi-label head given the labels $\lambda_1, \lambda_2, \lambda_3,$ and $\lambda_4$ (cf. Rapp, Loza Mencía, et al. (2018), Fig. 1). . . . .	20
2.	The peak relaxation lift function with parameters peak_label $m = 5,$ curvature $k = 1.2$ and max_lift $l = 1.5$ for a data set with $n = 10$ different labels . . . . .	21
3.	Sequence diagram of the flow of the implemented algorithm . . . . .	22
4.	Numeric Condition Finding . . . . .	27
5.	Nominal Condition Finding . . . . .	28
6.	Sparse Condition Finding . . . . .	30
7.	Average ranks of all tested configurations using the single-label head refinement . . . . .	38
	(a). data set emotions . . . . .	38
	(b). data set birds . . . . .	38
	(c). data set medical . . . . .	38
	(d). data set scene . . . . .	38
	(e). data set yeast . . . . .	38
8.	Average ranks of all tested configurations using the multi-label head refinement . . . . .	40
	(a). data set emotions . . . . .	40
	(b). data set birds . . . . .	40
	(c). data set medical . . . . .	40
	(d). data set scene . . . . .	40
	(e). data set yeast . . . . .	40

---

## List of Tables

---

1.	Example attributes and conditions . . . . .	12
2.	Example decision rule list with n rules . . . . .	14
3.	Example evaluation scores of single-label rules on the data set WEATHER with equal body . . . . .	21
4.	Relevant characteristics of the data sets used in the experiments . . . . .	35
5.	Tuned parameters and tested values. The peak_label parameter is chosen as the cardinality, rounded up to the next positive integer . . . . .	35
6.	Best performances all models learned. . . . .	37
7.	Performance of multi-label models learned from the data set birds . . . . .	42
8.	Performance of multi-label models learned from the data set emotions . . . . .	43
9.	Performance of multi-label models learned from the data set yeast . . . . .	45
10.	Performance of multi-label models learned from the data set medical . . . . .	46
11.	Performance of multi-label models learned from the data set scene . . . . .	48
12.	Performance of single-label models learned from the data set birds . . . . .	49
13.	Performance of single-label models learned from the data set emotions . . . . .	50
14.	Performance of single-label models learned from the data set yeast . . . . .	50
15.	Performance of single-label models learned from the data set medical . . . . .	50
16.	Performance of single-label models learned from the data set scene . . . . .	51
17.	Model characteristics of multi-label models learned from the data set birds . . . . .	52
18.	Model characteristics of multi-label models learned from the data set emotions . . . . .	53
19.	Model characteristics of multi-label models learned from the data set yeast . . . . .	55
20.	Model characteristics of multi-label models learned from the data set medical . . . . .	56
21.	Model characteristics of multi-label models learned from the data set scene . . . . .	58
22.	Model characteristics of single-label models learned from the data set birds . . . . .	59
23.	Model characteristics of single-label models learned from the data set emotions . . . . .	60
24.	Model characteristics of single-label models learned from the data set yeast . . . . .	60
25.	Model characteristics of single-label models learned from the data set medical . . . . .	60
26.	Model characteristics of single-label models learned from the data set scene . . . . .	61

---

# List of Algorithms

---

1. Algorithm SEPARATEANDCONQUER for iteratively learning rule lists for binary classification (cf. Fürnkranz (1999)) . . . . .	13
2. Algorithm INDUCERULES for inducing multi-label rule lists . . . . .	23
3. Algorithm INDUCEDEFAULTRULE for initializing the matrix $C_{default}$ and inducing the default rule	24
4. Algorithm INDUCERULE for finding the best rule on the remaining training data set . . . . .	25
5. Algorithm SHOULDCONTINUE for determining whether or not the algorithm should stop learning rules or continue . . . . .	25
6. Algorithm REFINERULE for finding and appending the best condition to a rule body . . . . .	26
7. Algorithm UPDATESEARCH for aggregating the confusion matrix $C_{covered}$ . . . . .	29
8. Algorithm RESETSEARCH for resetting the state of the condition search . . . . .	29
9. Algorithm FINDHEAD for finding and setting the best head for a rule . . . . .	31
10. Algorithm FINDSINGLELABELHEAD for finding and setting the best single-label head for a rule	31
11. Algorithm FINDMULTILABELHEAD for finding and setting the best multi-label head for a rule .	32
12. Algorithm EVALUATELABELPREDICTIONS for computing the evaluation scores for all labels based on matrices $C_{covered}$ and $C_{default}$ . . . . .	33

---

# 1. Introduction

---

This initial chapter gives an overview of the content and structure of this work. The first section briefly introduces the field of multi-label classification. After that, the objectives of this thesis and a short summary of each chapter are presented.

---

## 1.1. Multi-Label Classification

---

Classification is one of the most relevant applications of supervised machine learning. The classification problem may be divided into three domains; binary classification, multi-class classification and multi-label classification.

- **Binary classification:**  
Binary classification is the task of determining to which of two distinct classes an example belongs to. For example, determining whether an email is spam or not belongs to this category.
- **Multi-Class classification:**  
Multi-class classification is used when three or more classes exist and a given example shall be assigned to exactly one of those classes. For example, determining which single animal is depicted on a photograph belongs to this category.
- **Multi-Label classification:**  
Multi-label classification is used when any given example is assigned to any number of a set of classes or none of them. For example, determining which of a given set of topics a newspaper article is about belongs to this category.

---

## 1.2. Goal of the Thesis

---

This thesis implements an efficient multi-label rule learning algorithm which provides accurate predictions based on human readable rule lists, which assign labels to diverse input data. In order to achieve this, a separate-and-conquer approach for learning multi-label rules combined with the relaxed pruning approach proposed by Klein (2018) is implemented in this work. The influence of hyperparameter settings on different data sets is then analyzed in order to find optimal configurations for various environments. Even though some work related to implementing or tuning hyperparameters of multi-label head rule learners exists (cf. Rapp (2016); Rapp, Loza Mencía, et al. (2018); Rapp, Loza Mencía, et al. (2019); Klein (2018)), more extensive evaluation, especially for approaches using the relaxed pruning approach (see chapter 2.4.3) and adapted multi-label evaluation metrics (see chapter 2.3.3) is conducted in this work. Furthermore, a more

---

detailed implementation of the algorithm proposed by Rapp (2016) is introduced, including algorithmic improvements over a naive implementation of the base algorithm.

---

### 1.3. Organization of the Thesis

---

In this chapter, the contents and objectives of each chapter are briefly described.

- **Chapter 2:**  
This chapter introduces the formal definitions, notations and concepts which lay the foundation for the rest of this thesis. This includes multi-label classification, problem transformation methods, label dependencies, separate-and-conquer rule learning, different performance measures and averaging strategies. Furthermore, two properties of multi-label evaluation functions, decomposability and anti-monotonicity, are introduced.
- **Chapter 3:**  
In this chapter, the implemented rule learning algorithm is introduced. This includes an implementation of the rule induction process, refinement of rule conditions and the algorithms for head refinement.
- **Chapter 4:**  
In this chapter, the performance of the algorithm with different hyperparameter settings is examined. This includes a description of the experimental setup, tuned parameters, tested data sets, an analysis of the obtained results and multiple figures showing the performance of the implemented algorithm with different parameters. Furthermore, the characteristics of the obtained models are analyzed.
- **Chapter 5:**  
This final chapter includes a summary of the results and contributions of this work as well as a composition of possible future improvements of the implemented algorithm.

---

## 2. Fundamentals

---

This chapter introduces the basic concepts of multi-label classification and inductive rule learning. This includes formal definitions, notations and explanations which lay the foundation for the algorithm presented in this work.

---

### 2.1. Multi-Label Classification

---

Supervised machine learning is the task of learning a function, which will be called *model* in this work, that maps an input  $X$  to an output  $Y$  based on example input-output pairs, called the *training data set*  $T$  (Mohri et al., 2018).

$$T := \{(X_1, Y_1), \dots, (X_i, Y_i)\} \in \mathbb{X} \times \mathbb{Y}$$

The training data set consists of sets of values  $x$  corresponding to *attributes*  $A$ , each associated with an output  $Y$ . These sets are called *examples* in this work and each example  $X$  assigns values to a fixed number  $j$  of attributes  $A$ .

$$X_i := (x_1, \dots, x_j) \in \mathbb{X}, \text{ with } \mathbb{X} = A_1 \times \dots \times A_j$$

In this work, *nominal* and *numerical* attributes are considered. Values of nominal attributes are discrete, e.g. **red** or **green**, while the value of numerical attributes can be any number from a continuous value range (Janssen, 2012; Fürnkranz, 1999).

Classification is an instance of supervised machine learning and corresponds to the problem of identifying to which of a set of classes or *labels* a new example belongs, based on the training data set containing examples whose assigned labels are known (Alpaydin, 2010). That means in classification, the output  $Y$  of the model is one or several of a fixed set of classes. This set is called *label vector* in this work, in which each attribute  $y_i$  denotes the presence or absence of the corresponding label  $\lambda_i$ . The value 1 of a label  $y_i$  denotes the presence of the corresponding label  $\lambda_i$  and 0 denotes its absence (cf. Loza Mencía (2012); Loza Mencía and Janssen (2016)).

$$Y_i := (y_1, \dots, y_k) \in \{0, 1\}^k, \text{ with } k = |\mathbb{L}|$$

These classes will be called *labels* in this work and their corresponding domain is the finite *label space*  $\mathbb{L}$ .

$$\mathbb{L} = \{\lambda_1, \dots, \lambda_k\}$$

The learned model can be described as a classifier function that maps yet unknown examples to a *prediction*  $\hat{Y}$  (Loza Mencía and Janssen, 2016):

$$f : \begin{cases} \mathbb{X} \rightarrow \mathbb{Y} \\ X \mapsto \hat{Y} \end{cases}$$

---

### 2.1.1. Problem Transformation

Many approaches to multi-label classification include transforming the problem into multiple smaller single-label or regression problems which are then solved in a conventional way (Loza Mencía, 2012; Read et al., 2009). The most common of these *problem transformation methods* include:

- **Binary Relevance:** This method is based on training one binary classifier for each individual label to predict its presence or absence. By decomposing the multi-label problem into  $k$  independent binary subproblems, linear complexity with respect to the number of labels can be achieved using conventional binary classifiers (M.-L. Zhang, Li, et al., 2017). In order to predict the labels for an unseen example, the predictions of all classifiers have to be combined into one label vector. Because this method considers each label individually, dependencies between labels (see 2.1.2) cannot be exploited in order to improve classification performance (Read et al., 2009).

$$\hat{Y} = (f_1(X), \dots, f_k(X))$$

- **Pairwise Decomposition:** In this method, one binary classifier is trained for each pair of labels. Each classifier then predicts which of these two labels should be preferred over the other for each example (Loza Mencía, 2012). This decomposes the multi-label classification problem into  $\frac{n(n-1)}{2}$  binary subproblems. In order to predict the labels for an unseen example, the predictions of each classifier are interpreted as a vote for one of the corresponding labels. After aggregating these votes, the labels can be sorted by their relevance to the given example. The labels which are then included in the final multi-label prediction are all labels whose aggregated votes are above a defined threshold (Fürnkranz et al., 2008). A problem with this approach is the quadratic number of classifiers which have to be trained and evaluated in relation to the number of labels (Loza Mencía and Janssen, 2016).
- **Label Powerset:** In this method, each possible label combination which appears in the training data set is considered as a separate class (Read et al., 2009). Given  $k$  different labels exist,  $2^k$  different label combinations have to be considered in the worst case. This decomposes the multi-label classification problem into a multi-class classification task with up to  $2^k$  classes. In order to predict the labels for an unseen example, the label combination which is predicted by the multi-class classifier is used as the resulting multi-label prediction. However, this method tends to overfit for small data sets, as only label-combinations which appear in the training data set can be predicted by the resulting classifiers and the computational complexity of this approach can become a prohibitive bottleneck (Spolaôr et al., 2013).

### 2.1.2. Label Dependencies

In many data sets, implicit dependencies between labels exist and exploiting these dependencies may yield an enhanced performance (M.-L. Zhang and K. Zhang, 2010). For example, in a multi-label data set assigning topics to newspaper articles, a dependency between the topics **science** and **technology** might exist. Label dependencies can be divided into two classes; *conditional* and *unconditional* label dependencies (Dembczynski et al., 2012). If the dependencies do not depend on certain attributes in the data set, they are called *unconditional* or *global*. If they do depend on certain attributes of specific examples and thus only exist for a subset of the training data, they are called *conditional* or *local* (Loza Mencía, 2012). The dependency in the previously shown example is unconditional as the dependency exists globally in the data set. However, in a multi-label data set which assigns colors to animals based on their attributes, a dependency between the colors **black** and **yellow** might exist for insects but not for other animals such as birds. Therefore, the dependency would only be local.

attribute	condition type	possible values	examples
weather	nominal	sunny, rainy, foggy	sunny, $\neg$ foggy
temperature	numeric	$[0, \infty)$	temperature $> 85.0$ , temperature $\leq 70$

Table 1.: Example attributes and conditions

## 2.2. Inductive Rule Learning

In order to derive a model from the training data set, the implemented algorithm learns individual *rules*. Each rule consist of a *body* and a *head*. The body of the rule consists of attribute-value tests, called *conditions* and the head of the rule provides predictions for a single or multiple labels.

$$\text{body} \rightarrow \text{head}$$

A rule *covers* an example if all conditions in the body are satisfied by the attribute values of the example. In this work, only propositional, conjunctive rules, whose conditions are concatenated using logical AND ( $\wedge$ ) operations are considered. For nominal attributes, equality ( $=$ ) and inequality ( $\neq$ ) tests are used, while relational tests ( $\leq$ , and  $>$ ) are used for numerical attributes. Equality and inequality checks are abbreviated using the notation  $x$  or  $\neg x$  respectively in this work. Label predictions will be denoted as  $\hat{y}$  for the presence of a label and  $\neg\hat{y}$  for the absence of a label respectively.

For example the rule

$$\text{temperature} > 10 \wedge \text{relative\_humidity} \leq 0.5 \rightarrow \neg\text{fog}, \neg\text{snow}, \text{sunny}$$

implies that if the **temperature** is greater than 10 and the **relative\_humidity** is below or equal to 0.5, the labels **fog** and **snow** are considered to be irrelevant, while the label **sunny** is considered to be relevant. As rules can easily be comprehended by humans, rules and rule lists provide a way to describe multi-label classification models interpretable by humans. Table 1 shows example attributes and conditions.

### 2.2.1. Rule Types

Multi-label rules can be divided into *single-label head rules* and *multi-label head rules*. While single-label head rules only contain one label in the rule head, multi-label head rules predict multiple labels at once (Rapp, 2016). Therefore, multi-label head rules can express dependencies between labels (see chapter 2.1.2). For example, the rule head " $\rightarrow \neg\text{moonshine}, \text{sunshine}$ " might indicate that the labels **sunshine** and **moonshine** are globally or locally mutually exclusive. Rule heads which predict a value for all labels in the data set will be called *full* rule heads in this work.

### 2.2.2. Separate-and-Conquer Rule Learning

An often used approach to learning rules is *separate-and-conquer* rule learning. In this approach, the algorithm starts with an empty list of rules and adds rules successively to this list (Fürnkranz, 1999). First, the best

rule according to an evaluation function is selected and added to the list. After that, all examples, which the newly learned rule covers, are removed from the training data set. Algorithm 1 shows the basic structure of separate-and-conquer algorithms.

---

**Algorithm 1.** Algorithm SEPARATEANDCONQUER for iteratively learning rule lists for binary classification (cf. Fürnkranz (1999))

---

**Input:** Training data set  $T$

**Output:** Learned rule list RuleList

```

1: procedure SEPARATEANDCONQUER
2:   RuleList =  $\emptyset$ 
3:   while GETPOSITIVEEXAMPLES( $T$ )  $\neq \emptyset$  do           ▷ Learn rules until no positive examples remain
4:     BestRule = FINDBESTRULE( $T$ )
5:      $T = T \setminus$  GETCOVERED( $T$ , BestRule)           ▷ Remove covered examples from the training set
6:     RuleList = RuleList  $\cup$  BestRule
7:   end while
8:   return RuleList
9: end procedure

```

---

This algorithm guarantees *completeness* as it only terminates after all positive examples are covered. However, in most cases this would lead to overfitting the data set (Loza Mencía and Janssen, 2016; Fürnkranz, 1999). In order to avoid that, the parameter `max_rules` is used in the experiments in chapter 4 to stop the learner from overfitting by learning more than `max_rules` rules.

### 2.2.3. Decision Lists

The rule lists returned by separate-and-conquer algorithms are called *decision lists*. These decision lists are ordered sets containing the learned rules.

$$\mathbb{D} = (r_1, r_2, \dots)$$

When an example is classified with a single-label decision list for multi-class classification, the first rule which covers the example determines the predicted class.

In multi-label classification, the concept of decision lists has to be adapted in order to support predictions for multiple labels in one evaluation. In contrast to conventional decision lists, the classification process is not stopped when an example is covered by a rule, but continues until predictions for all labels exist. As predictions of rules evaluated in the decision list cannot be revoked by other rules later, only predictions for yet unpredicted labels are considered. In order to guarantee that all labels are predicted at the end of the decision list, multi-label decision lists contain a *default rule*.

### 2.2.4. Default Rule

The default rule is the last rule of a decision list and makes a prediction for every yet unpredicted label. In order to achieve this, the default rule has a full label head and predicts the presence or absence of all possible labels. Consequently, rules other than the default rule only predict labels opposite to the default rule as abstaining from a prediction for that label corresponds to predicting the label predicted by the default rule. In

#	Rule
1	$humidity > 82.5 \wedge temperature > 82.5 \wedge outlook = sunny \rightarrow \neg play, dontplay$
2	$humidity > 88 \wedge temperature > 76 \wedge outlook = sunny \rightarrow \neg play, dontplay$
3	$temperature \leq 68.5 \wedge outlook \neq rainy \rightarrow playmaybe$
...	...
n	$\emptyset \rightarrow \mathbf{play = 1, dontplay = 0, playmaybe = 0}$

Table 2.: Example decision rule list with n rules

the experiments in chapter 4, the default rule is chosen to predict the majority class, i.e., the most frequent value in the training data set, for each label. Hence, models with only one learned rule, the default rule, provide a good baseline model for comparison with further extended models. Table 2 shows an example decision list with the default rule as the last rule of the list.

## 2.3. Evaluation Metrics

In order to assess the quality of multi-label predictions, evaluation functions assigning a single score to a prediction given by a multi-label classification model are required. Those evaluation metrics can be divided into bipartition and ranking evaluation measures. However, this chapter only introduces bipartition evaluation measures, as ranking evaluation measures are not considered in this work. First, this chapter will introduce the classic notation for confusion matrices as well as an adapted notation proposed by Hefter (2020). After that, multiple selected evaluation metrics are presented as well as adapted versions for the multi-label classification setting. Chapter 2.3.4 describes different aggregation and averaging strategies used to combine the scores for multiple predictions into a single score for a rule. Finally, two properties of multi-label evaluation functions defined by Rapp (2016), which will be used to reduce the search complexity for finding multi-label head rules, are presented.

### 2.3.1. Bipartition Evaluation

Bipartition evaluation metrics evaluate a model prediction based on the comparisons of the true label vector to the label vector predicted by the model. These differences can be visualized in a *confusion matrix C*:

	predicted	not predicted
relevant	TP	FN
irrelevant	FP	TN

The entries of the matrix correspond to the results of the comparison between the true label vector and the predicted label vector:

- True positives (TP): The label is present in the true and the predicted label vector
- False negatives (FN): The label is present in the true label vector, but not in the predicted label vector
- False positives (FP): The label is not present in the true label vector, but is present in the predicted label vector
- True negatives (TN): The label is neither present in the true nor in the predicted label vector

### 2.3.2. Multi-Label Evaluation

In order to evaluate multi-label rule predictions, metrics other than the conventional metrics for binary and multi-class classifications are required (Loza Mencía, 2012). Hence, various of those conventional metrics have been adapted to evaluate the quality of multi-label classifications. In this work, an adapted notation for the evaluation of rule list model predictions is used as proposed by Hefter (2020). In contrast to the binary setting, an additional dimension of entries is required for the confusion matrix in the multi-label rule setting:

- **covered/uncovered (C/U)**: whether the the example is covered by the rule or not
- **relevant/irrelevant (R/I)**: whether the label is true (1) or false (0) according to the true label vector
- **predicted positive/predicted negative (P/N)**: whether the rule predicted the label to be true (1) or false (0)

This results in a different confusion matrix for the multi-label setting:

	$y$	$\hat{y}$	name
covered	0	0	$C_{in}$
covered	0	1	$C_{ip}$
covered	1	0	$C_{rn}$
covered	1	1	$C_{rp}$
uncovered	0	0	$U_{in}$
uncovered	0	1	$U_{ip}$
uncovered	1	0	$U_{rn}$
uncovered	1	1	$U_{rp}$

Other approaches for multi-label evaluation are possible as well. For example, all eight entries of the extended confusion matrix may be mapped to the four entries of the classic notation for confusion matrices. However, this leads to a loss of information. For example, if all correct predictions ( $y = \hat{y}$ ) are classified as true positives, evaluation metrics cannot distinguish between a correctly predicted positive label and a correctly predicted negative label.

Using the confusion matrices above, classic metrics and metrics adapted for the multi-label setting can be defined.

### 2.3.3. Selected Metrics

The functions used to evaluate rules or predictions will be called *metrics* in this work. As these functions provide an approximation for the quality of a rule, they are also often called *heuristics* (Janssen, 2012). In this chapter, the metrics used or mentioned in this work are introduced. For each metric, the classic definition is presented as well as a version adapted for the multi-label setting, if required in the remainder of this work. The following shorthand notations are used in the definitions below:

$$\begin{aligned}
 C_{correct} &= C_{rp} + C_{in} \\
 C_{incorrect} &= C_{rn} + C_{ip} \\
 C &= C_{correct} + C_{incorrect} \\
 U_{correct} &= U_{rp} + U_{in} \\
 U_{incorrect} &= U_{rn} + U_{ip} \\
 U &= C_{correct} + U_{incorrect} \\
 P &= tp + fp \\
 N &= tn + fn
 \end{aligned} \tag{2.1}$$

- **Precision:**

Precision computes the fraction of relevant labels among all predicted labels and is defined as:

$$\delta_{prec} = \frac{tp}{tp + fp} \tag{2.2}$$

The version adapted for the multi-label setting is defined as the fraction of correctly predicted labels among all covered labels:

$$\delta_{prec}^{adapted} = \frac{C_{correct}}{C} \tag{2.3}$$

- **Hamming accuracy:**

The Hamming loss is a common metric for binary and multi-class classification and is defined as follows:

$$\delta_{hammloss} = \frac{fp + fn}{tp + fp + tn + fn} \tag{2.4}$$

In this work, a gain metric constructed from Hamming loss, the Hamming accuracy, is used. It is defined as  $1 - \delta_{hammloss}$  and corresponds to the percentage of correctly classified labels among all labels:

$$\delta_{hammacc} = \frac{tp + tn}{tp + fp + tn + fn} \tag{2.5}$$

- **Recall:**

Recall computes the fraction of predicted labels among all relevant labels and is defined as:

$$\delta_{rec} = \frac{tp}{tp + fn} \tag{2.6}$$

The adapted version computes the percentage of covered labels among all labels which would be correctly classified by the rule to evaluate:

$$\delta_{rec}^{adapted} = \frac{C_{correct}}{C_{correct} + U_{correct}} \quad (2.7)$$

- **F-Measure:**

The F-measure computes the harmonic mean between precision and recall and is defined as:

$$\delta_F = \frac{(\beta^2 + 1) \cdot \delta_{prec} \cdot \delta_{rec}}{\beta^2 \cdot \delta_{prec} + \delta_{rec}} \quad (2.8)$$

The positive, real parameter  $\beta$  controls how much more important recall is considered than precision. This means if  $\beta > 1$ , recall is considered more important than precision, if  $\beta < 1$ , precision is considered more important than recall and if  $\beta = 1$ , they are both considered equally important. The metric with this special case ( $\beta = 1$ ) is often referred to as "F1-measure" and is defined as:

$$\delta_{F1} = \frac{2 \cdot \delta_{prec} \cdot \delta_{rec}}{\delta_{prec} + \delta_{rec}} \quad (2.9)$$

- **M-Estimate**

The M-estimate can be interpreted as a trade-off between weighted relative accuracy and precision and is defined as:

$$\delta_m = \frac{tp + m * \frac{P}{P+N}}{tp + fp + m}, \text{ with } m \geq 0 \quad (2.10)$$

The version adapted for the multi-label setting is defined as follows:

$$\delta_m^{adapted} = \frac{C_{correct} + (m * \frac{C_{correct} + U_{correct}}{(C+U)})}{C + m}, \text{ with } m \geq 0 \quad (2.11)$$

In the experiments in evaluation chapter 4, the evaluation metric M-estimate is used with different values for the parameter  $m$ .

### 2.3.4. Aggregation and Averaging

In order to derive a metric for multi-label models from the evaluations of multi-label predictions using bipartition functions, a single, comparable score needs to be calculated. One of the most commonly used approaches to do this is the averaging of these binary classification metrics (Koyejo et al., 2015). *Micro-averaging* and *macro-averaging* are two well-known averaging strategies.

Two aggregation operators are required in order to formally define the different averaging strategies. Let  $C_i$ ,  $i = 1, \dots, n$  be a sequence of confusion matrices and  $\oplus$  the cell-wise addition of confusion matrices.

$$\sum_{i=1}^n C_i := C_1 \oplus \dots \oplus C_n$$

$$\text{avg}_{i=1}^n C_i := \frac{1}{n} \sum_{i=1}^n C_i$$

Now, the different averaging strategies can be defined. Since it is possible to iterate over two different dimensions, examples and labels, before or after applying the evaluation function, six combinations of aggregation exist. However, only four of these strategies are mathematically distinct (Loza Mencía, 2012). The following definitions are based on Koyejo et al. (2015) and Loza Mencía (2012). Let  $i$  iterate over all labels  $\lambda_1, \dots, \lambda_n$  and  $j$  over all of examples  $X_1, \dots, X_m$  of the data set.

- **(Label and Example-based) Micro-Averaging:** First, the atomic confusion matrices of all labels and all examples are summed up to one single global confusion matrix. The evaluation function  $\delta$  is then applied to that confusion matrix in order to calculate the final single score.

$$\delta \circ \sum_i \circ \sum_j = \delta \circ \sum_j \circ \sum_i \quad (2.12)$$

- **Example-Based (Macro-)Averaging:** First, the confusion matrix for each example is calculated by adding up the atomic confusion matrix of each label of that example. Then, one score is calculated for each example by applying the evaluation function  $\delta$  to each of these confusion matrices. Finally, a final, single score is calculated by computing the arithmetic mean of these scores.

$$avg_j \circ \delta \circ \sum_i \quad (2.13)$$

- **Label-Based (Macro-)Averaging:** First, one confusion matrix for each label is calculated by adding up the atomic confusion matrix of this label for each example. Then, one score is calculated for each label by applying the evaluation function  $\delta$  to each of these confusion matrices. Finally, a final, single score is calculated by computing the arithmetic mean of these scores.

$$avg_i \circ \delta \circ \sum_j \quad (2.14)$$

- **(Label and Example-Based) Macro-Averaging:** First, the evaluation function is applied to each atomic confusion matrix of each label and example. Then, the calculated scores are averaged in order to obtain one final score.

$$avg_i \circ avg_j \circ \delta = avg_j \circ avg_i \circ h \quad (2.15)$$

In this work, only label-based averaging is implemented as it fulfills the conditions required for pruning based on decomposability (see chapter 2.4.2) independently of the metric used for evaluation. The following chapter describes the conditions and advantages of different pruning strategies.

---

## 2.4. Pruning the Label Space

---

A very basic approach to finding the best head for a given rule body would be evaluating all possible heads and selecting the head which lead to the best performing rule as the best head for the rule. However, an exhaustive search of the label space  $\mathbb{L}$  would be infeasible in most cases because its size grows exponentially in relation to a possibly large number of labels (Loza Mencía, 2012). This chapter introduces two properties of multi-label evaluation metrics as defined by Rapp (2016), which can be exploited in order to limit the amount of evaluations while still guaranteeing the best solution. The following two chapters are based on Rapp (2016) and Rapp, Loza Mencía, et al. (2018).

### 2.4.1. Anti-Monotonicity

Anti-monotonous evaluation functions allow effective pruning of the required search space by guaranteeing that certain combinations of labels cannot reach the maximum performance, if at least one of its subsets does neither. This is true for all evaluation functions which guarantee that if a rule heads performance has decreased by adding a label to the rule head, the maximum performance reached before cannot be reached again by adding more labels to the rule head. Given a specific averaging and evaluation strategy, a multi-label evaluation function  $\delta$  is considered anti-monotonous, if the following conditions are met:

- If the performance of a multi-label head rules has decreased because a label has been added to the rule head, the maximum performance  $h_{max}$  cannot be reached anymore by adding more labels to the rule head:

$$\hat{Y}_p \subset \hat{Y}_s \wedge \delta(B \rightarrow \hat{Y}_s, T) < \delta(B \rightarrow \hat{Y}_p, T) \Rightarrow \delta(B \rightarrow \hat{Y}, T) < h_{max}, \forall \hat{Y}_a (\hat{Y}_p \subset \hat{Y}_a) \quad (2.16)$$

- $\delta$  must not be monotonous according to Definition 2.17.

As monotonous evaluation functions are not considered in this work, this chapter will only contain the formal definition of monotonicity required for the definitions of anti-monotonicity and decomposability. Given a specific averaging and evaluation strategy, a multi-label evaluation function  $\delta$  is considered monotonous if adding a new label to an existing rule head never causes that rules performance to decrease:

$$\delta(B \rightarrow \hat{Y}_a, T) \geq \delta(B \rightarrow \hat{Y}_p, T), \forall \hat{Y}_a (\hat{Y}_p \subset \hat{Y}_a) \quad (2.17)$$

### 2.4.2. Decomposability

Decomposability is a stronger property and implies anti-monotonicity. An evaluation function is considered to be decomposable, if combining multiple single-label rules with different performances to a multi-label head rule always yields a rule with worse performance than the best performing single-label head rule. Only multi-label head rules which are composed of best performing single-label head rules can reach the best performance as well. Therefore, only single-label heads reaching the maximum performance need to be considered and combined after all possible single-label heads have been evaluated. Formally, a multi-label evaluation function  $\delta$  is considered decomposable, if:

- If a single-label rule  $B \rightarrow \hat{y}$  corresponding to one predicted label attribute  $\hat{y}$  in the head of a multi-label head rule  $B \rightarrow \hat{Y}$  does not reach the best possible performance  $h_{max}$ , the multi-label rule itself does not reach that maximum performance either, and vice versa:

$$\exists \hat{y} \in \hat{Y} (\delta(B \rightarrow \hat{y}, T) < h_{max}) \iff \delta(B \rightarrow \hat{Y}, T) < h_{max}$$

- If each single-label rule  $B \rightarrow \hat{y}$  corresponding to the predicted label attributes  $\hat{y}$  in the head of a multi-label head rule  $B \rightarrow \hat{Y}$  reaches the best possible performance  $h_{max}$ , the multi-label rule itself reaches exactly that maximum performance as well, and vice versa:

$$\forall \hat{y} \in \hat{Y} (\delta(B \rightarrow \hat{y}, T) = h_{max}, ) \iff \delta(B \rightarrow \hat{Y}, T) = h_{max}$$

- $\delta$  must not be monotonous according to Definition 2.17.

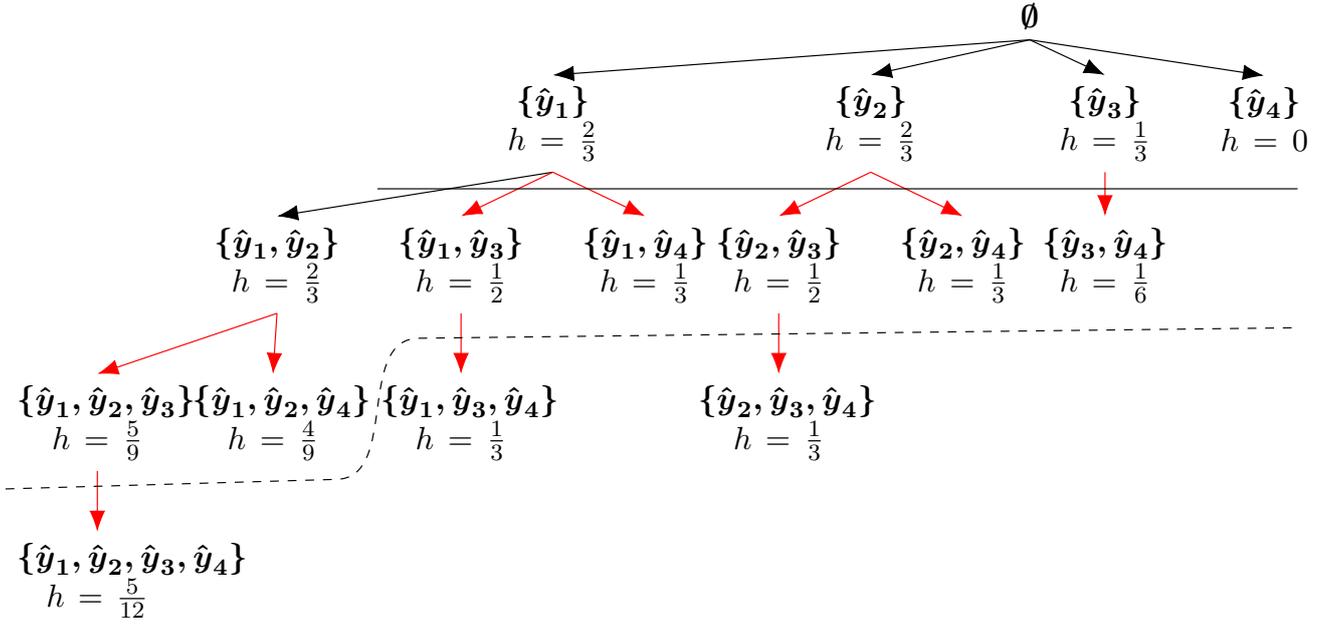


Figure 1.: Prunable search for the best multi-label head given the labels  $\lambda_1, \lambda_2, \lambda_3,$  and  $\lambda_4$  (cf. Rapp, Loza Mencía, et al. (2018), Fig. 1).

Instead of performing a deep search through the entire label space, the property of decomposability of the evaluation function can be used in order to fundamentally reduce the computational cost of determining the best head for a rule. Instead of performing  $2^n$  evaluations for  $n$  labels, the number of evaluations can be reduced to  $2n$ . For each possible label, only two single-label rules need to be evaluated, predicting the presence or the absence of that label. Figure 1 shows how the label search space can be pruned according to decomposability or anti-monotonicity. All label combinations under the solid line can be pruned according to decomposability and all label combinations below the dashed line can be pruned according to anti-monotonicity.

Because label-wise averaging, as described in chapter 2.3.4, computes the arithmetic mean of all performances, it fulfills the conditions of decomposability independently of the metric used for evaluation. Therefore, the M-estimate metric used in the conducted experiments is decomposable, when it is used in combination with the label-wise averaging strategy.

### 2.4.3. Relaxed Pruning

When pruning the search through the label space according to decomposability, only single-label rules or multi-label rules, whose heads only contain labels for which the corresponding single-label rules evaluate to the same evaluation score, are learned. Because the evaluation scores of these single-label rules often differ in practice, this results in learning mostly single-label rules missing the possibility of exploiting label dependencies (Klein, 2018). The example in table 3 shows that label dependencies may be missed, even if the evaluation scores of the corresponding single-label rules only differ slightly.

In order to learn multi-label head rules possibly revealing label dependencies, Klein (2018) proposed relaxing the pruning constraints using Relaxation Lift Functions. Those relaxation lift functions allow the learner to learn multi-label rules even if the evaluation score of the corresponding single-label rules slightly differ. To

h	Rule
0.615	temperature $\leq 66.5 \wedge$ outlook = <i>rainy</i> $\rightarrow$ dontplay = 1
0.612	temperature $\leq 66.5 \wedge$ outlook = <i>rainy</i> $\rightarrow$ play = 0

Table 3.: Example evaluation scores of single-label rules on the data set WEATHER with equal body

calculate the lifted heuristic value of a multi-label head rule, the unlifted heuristic value is multiplied by the lift calculated by the relaxation lift function  $\rho$ . This lifted value is then used as the heuristic value for rule evaluation. In order to reinforce learning multi-label head rules, the function used as the relaxation lift function has to reward rules with more labels in the head more than rules with less labels in the head up to a certain point. Klein (2018) proposed and evaluated multiple different relaxation lift functions and concluded that the peak relaxation lift function reached the best performance in many cases. The implementation presented in this this work uses the peak relaxation lift function as defined in Equation 2.18 (cf. Klein (2018), Equation 5 and 6).

$$\rho_{peak}(x) = \begin{cases} 1 + \left(\frac{x-1}{m-1}\right)^{1/k} * (l-1) & \text{if } x \leq m \\ 1 + \left(\frac{n-x}{n-m}\right)^{1/k} * (l-1) & \text{if } x > m \end{cases} \quad (2.18)$$

Figure 2 shows a parameterized example of the peak relaxation lift function with the parameters  $m = 5$ ,  $k = 1.2$  and  $l = 1.4$  for a data set with  $n = 10$  different labels.

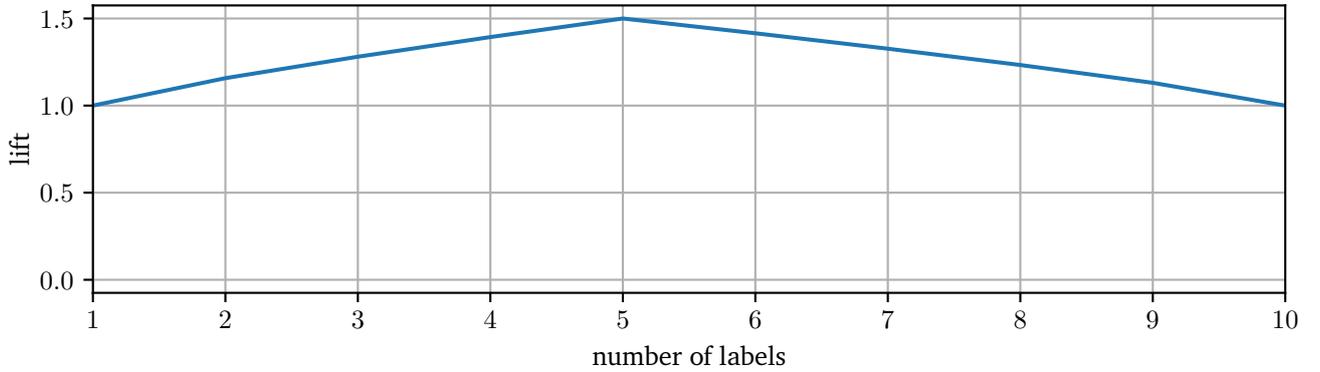


Figure 2.: The peak relaxation lift function with parameters peak\_label  $m = 5$ , curvature  $k = 1.2$  and max\_lift  $l = 1.5$  for a data set with  $n = 10$  different labels

### 3. Learning Multi-Label Rules

This chapter introduces the algorithm implemented in this work. It is based on the BOOMER framework (Rapp, Mencía, et al., 2020) and thus, it shares its underlying structure. The individual parts of the algorithm are described in the following chapters. Due to the modular structure of the framework, the specific parts of the algorithm can basically be interchanged with different implementations, resulting in other rule learning algorithms. Figure 3 shows a sequence diagram of the flow of the implemented algorithm.

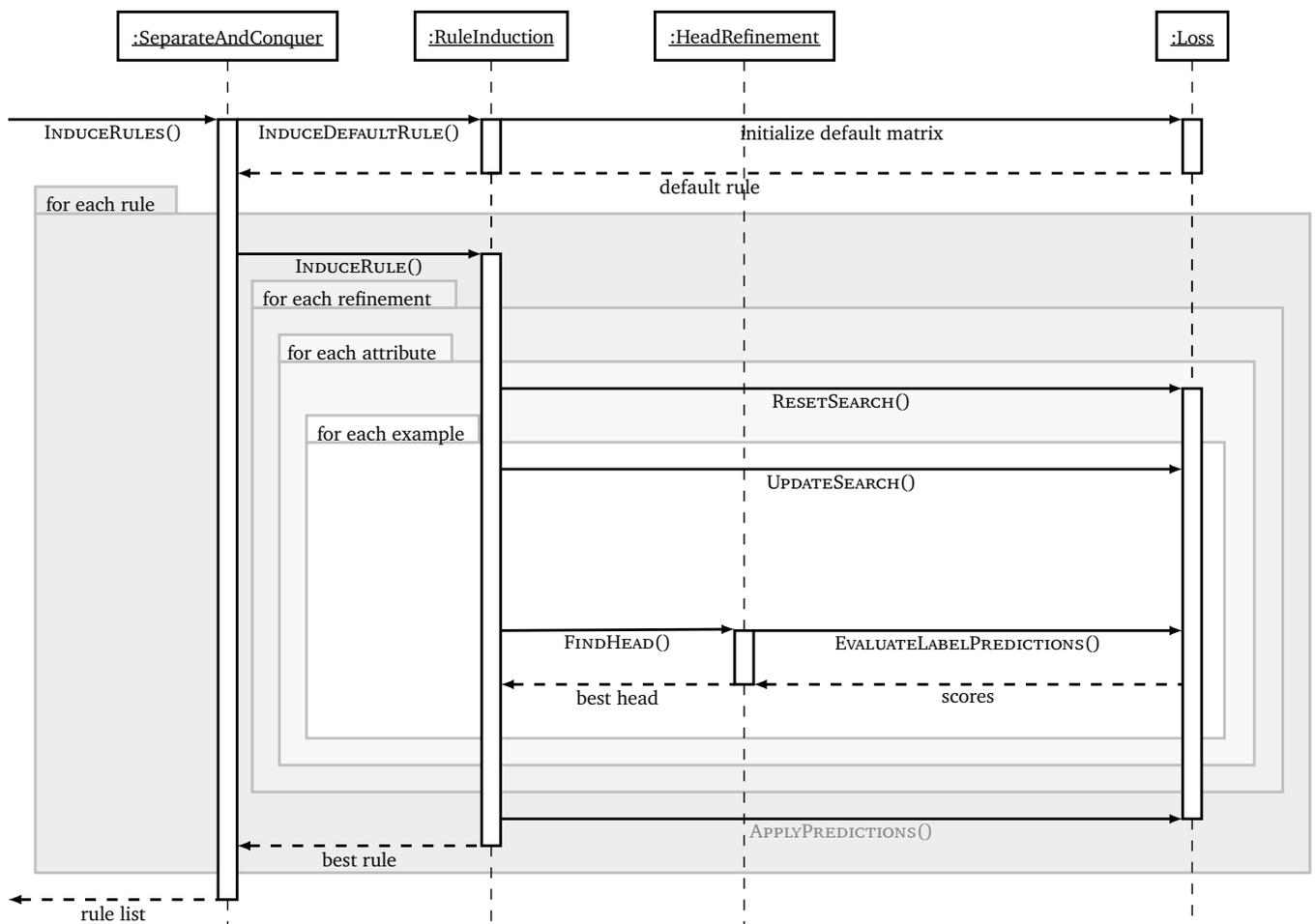


Figure 3.: Sequence diagram of the flow of the implemented algorithm

---

## 3.1. Rule Induction

---

Algorithm 2 INDUCERULES is the main routine of the algorithm and is responsible for combining induced rules to the final rule list. Even though the default rule induced by algorithm 3 INDUCEDEFAULTRULE is learned first, it is appended as the last element of the decision list. The reason for this is that the default rule is required for the condition refinement and must be initialized first. The algorithm continues learning rules until a stopping criterion defined in algorithm 5 SHOULDCONTINUE is satisfied.

---

**Algorithm 2.** Algorithm INDUCERULES for inducing multi-label rule lists

---

**Input:** Training data set T

**Output:** Learned rule list RuleList

```
1: procedure INDUCERULES
2:   RuleList =  $\emptyset$ 
3:    $r_{default}$  = INDUCEDEFAULTRULE(T)
4:   while SHOULDCONTINUE(RULELIST) do                                 $\triangleright$  Learn rules until stopping criterion is satisfied
5:      $r$  = INDUCERULE(T)
6:     RuleList = RuleList  $\cup$   $r$ 
7:   end while
8:   RuleList = RuleList  $\cup$   $r_{default}$ 
9:   return RuleList
10: end procedure
```

---

Algorithm 3 INDUCEDEFAULTRULE induces the default rule based on the majority labels as a prediction and initializes the matrix  $C_{default}$ . This matrix contains the label-wise confusion matrices which would result from predicting all examples according to the default rule. Matrix  $C_{default}$  will be used in algorithm 6 REFINERULE and explained in the corresponding chapter. The operator ++ is the increment operator, which increments the value of the preceding operand by 1.

---

**Algorithm 3.** Algorithm INDUCEDEFAULTRULE for initializing the matrix  $C_{default}$  and inducing the default rule

---

**Input:** Training data set  $T$

**Output:** Induces default rule  $r_{default}$

```

1: procedure INDUCEDEFAULTRULE
2:    $C_{default} = (0)^{|\mathbb{L}| \times 4}$  ▷  $C_{default}$  is initialized as an matrix of the shape  $|\mathbb{L}| \times 4$ 
3:   Let  $\hat{y}_i, \dots, \hat{y}_{|\mathbb{L}|}$  be the majority labels as a prediction
4:   for  $(X, Y) \in T$  do
5:     for  $\lambda_i \in \mathbb{L}$  do
6:       if  $\hat{y}_i = 1$  then
7:         if  $y_i = 1$  then
8:            $C_{default}^{TP} ++$ 
9:         else
10:           $C_{default}^{FP} ++$ 
11:        end if
12:      else
13:        if  $y_i = 1$  then
14:           $C_{default}^{FN} ++$ 
15:        else
16:           $C_{default}^{TN} ++$ 
17:        end if
18:      end if
19:    end for
20:  end for
21:   $r_{default} = \emptyset \rightarrow \hat{y}_i, \dots, \hat{y}_{|\mathbb{L}|}$ 
22:  return  $r_{default}$ 
23: end procedure

```

---

Algorithm 4 INDUCERULE finds the best rule for the currently remaining training set. For this, a greedy top-down search is used, starting with the most generic rule with an empty body, which is then successively specialized. In order to do this, conditions are appended to the body, making the rule more strict until it is not possible to improve it anymore. After each iteration, the remaining training data set is reduced to all examples which are covered by the induced rule. This is done using the subroutine GETCOVEREDEXAMPLES, which returns the subset of all examples which are covered by the induced rule  $r$ . This guarantees, that only examples which satisfy the previous conditions are considered in the condition search. The algorithm stops as soon as the evaluation score of the rule returned by REFINERULE is not better than the score of the best rule found yet anymore.

---

**Algorithm 4.** Algorithm INDUCERULE for finding the best rule on the remaining training data set

---

**Input:** Training data set  $T$ , evaluation function  $\delta$

**Output:** The best global rule found  $r_{best}$

```
1: procedure INDUCERULE
2:    $r_{best} = \emptyset$ 
3:    $r_{best}.h = -\infty$ 
4:   improved = true
5:   while improved do                                     ▷ While improvement is possible
6:      $r = \text{REFINERULE}(T, r_{best}, \delta)$ 
7:     if  $r.h > r_{best}.h$  then
8:        $r_{best} = r$ 
9:     else
10:      improved = false
11:    end if
12:     $T = \text{GETCOVEREDEXAMPLES}(r, T)$     ▷ Only examples which satisfy the previous conditions are
    considered for the next condition refinement
13:  end while
14:  return  $r_{best}$ 
15: end procedure
```

---

Algorithm 5 SHOULDCONTINUE determines whether or not the algorithm should stop learning rules or continue after a rule was learned. Even though many stopping criteria could be used here, only a simple stopping criterion is implemented in this work, leaving the testing of other stopping criteria for future work. The default stopping criterion for separate-and-conquer algorithms is that all positive examples in the training data set are covered (Fürnkranz, 1999). However, in this algorithm, an alternative stopping criterion adapted to the multi-label setting is implemented. The algorithm stops as soon as either the number of learned rules is greater than or equal to the parameter  $max\_rules$  or all labels have been covered by a rule, including the default rule. The variable  $SumUncoveredLabels$  corresponds to the number of labels which are neither predicted correctly by the default rule nor predicted by any rule covering the corresponding example.

---

**Algorithm 5.** Algorithm SHOULDCONTINUE for determining whether or not the algorithm should stop learning rules or continue

---

**Input:** Number of uncovered labels  $SumUncoveredLabels$ , rule list  $RuleList$ , maximum number of rules  $max\_rules$

**Output:** Whether or not the algorithm should continue learning rules

```
1: procedure SHOULDCONTINUE
2:   if  $SumUncoveredLabels = 0$  then
3:     return false
4:   else if  $|RuleList| \geq max\_rules$  then
5:     return false
6:   else
7:     return true
8:   end if
9: end procedure
```

---

---

## 3.2. Refining Rule Conditions

---

In order to choose the best refinement for a rule body, all possible refinements have to be evaluated. The algorithm 6 `REFINERULE`, called by algorithm 4 `INDUCERULE`, tries to find the best possible condition by exhaustively testing all possible attribute conditions. This is done efficiently by iterating through a sorted *AttributeValues* array of tuples (*value, index*) corresponding to the attribute values of each example in the training data set and its index (cf. Mehta et al. (1996); Shafer et al. (2000)). The `ARGSORT` subroutine sorts the array of attribute values and returns an array of tuples (*index, value*) containing the sorted attribute values and their corresponding indices in the training data set.

---

**Algorithm 6.** Algorithm `REFINERULE` for finding and appending the best condition to a rule body

---

**Input:** Training data set  $T$ , evaluation function  $\delta$ , rule  $r$

**Output:** The refined rule  $r_{best}$

```
1: procedure REFINERULE
2:    $r_{best} = r$ 
3:   RESETSEARCH()
4:   for each attribute  $A_i$  do
5:     Let AttributeValues be the array of all values assigned to attribute  $A_i$  in training data set  $T$ 
6:     AttributeValuessorted = ARGSORT(AttributeValues)
7:     for (value, index)  $\in$  AttributeValuessorted do
8:       UPDATESEARCH( $T[index]$ )
9:       if  $A_i$  is nominal then
10:         $r_{refined}.body = r.body \cup \lambda_i = value$ 
11:       else
12:         $r_{refined}.body = r.body \cup \lambda_i \leq value$ 
13:       end if
14:        $r = \text{FINDHEAD}(r, \rho, \text{false})$  ▷ Find the best head for covered examples
15:       if  $r.h > r_{best}.h$  then
16:         $r_{best} = r_{refined}$ 
17:       end if
18:       if  $A_i$  is nominal then
19:         $r_{refined}.body = r.body \cup \lambda_i \neq value$ 
20:       else
21:         $r_{refined}.body = r.body \cup \lambda_i > value$ 
22:       end if
23:        $r = \text{FINDHEAD}(r, \rho, \text{true})$  ▷ Find the best head for uncovered examples
24:       if  $r.h > r_{best}.h$  then
25:         $r_{best} = r_{refined}$ 
26:       end if
27:       if  $A_i$  is nominal then
28:        RESETSEARCH() ▷ Reset search as examples before are not covered by the next condition
29:       end if
30:     end for
31:   end for
32:   return  $r_{best}$ 
33: end procedure
```

---

For every one of these tuples, two conditions are then evaluated in each iteration. The operators for these conditions are the relational tests ( $\leq$  and  $>$ ) for numerical attributes and the equality tests ( $=$  and  $\neq$ ) for nominal attributes respectively. The procedure `UPDATESEARCH` updates the state of the condition search, keeping track of confusion matrices required to evaluate label predictions of rules with the current conditions. For each condition, the best head is determined by algorithm 9 `FINDHEAD`. The rule with the highest evaluation score  $h$  is then returned as the refined rule.

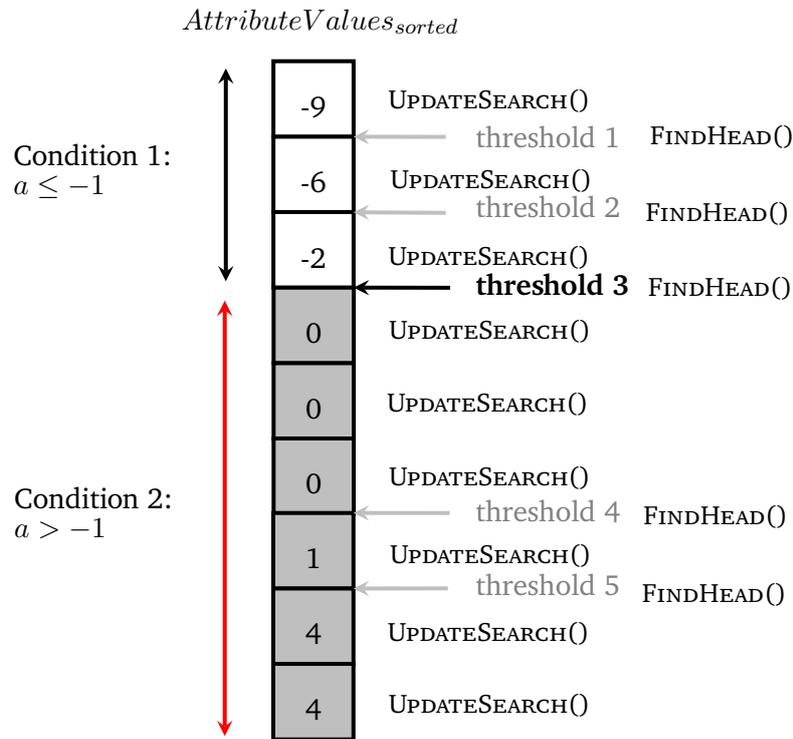


Figure 4.: Numeric Condition Finding

For nominal attributes, the algorithm needs to behave differently. Instead of relational tests, an equality and an inequality test has to be appended as a possible condition. Furthermore, the state of the condition search is reset after every example, as previous examples shall not count as covered for the upcoming refinement. The reason for this is that the equality test  $=$  is not satisfied for previous examples with other nominal values. Thus, the algorithm `RESETSEARCH` needs to be called after each iteration to reset the state of the condition search.

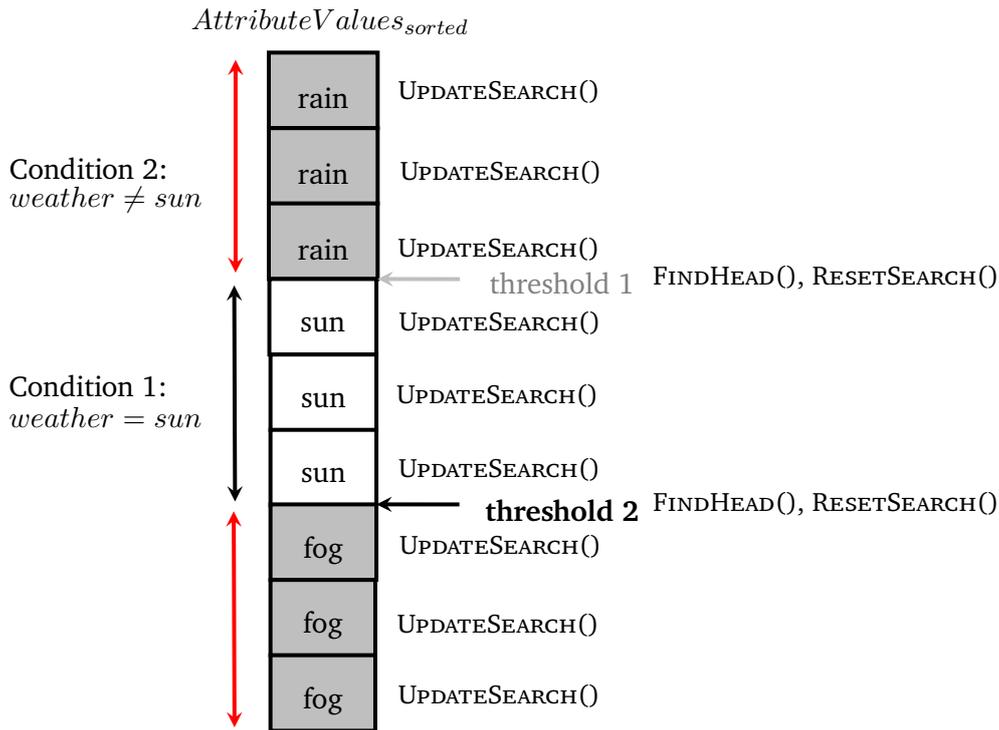


Figure 5.: Nominal Condition Finding

### 3.2.1. Aggregating Confusion Matrices

Algorithm 7 `UPDATESEARCH` is called by algorithm 6 `REFINERULE` and is responsible for aggregating the confusion matrix  $C_{covered}$  by comparing the labels which would be predicted by a rule with the true label vectors. It is called once for every example which will be covered by upcoming conditions. The matrix  $C_{covered}$  corresponds to the confusion matrix for all labels of all examples which assign a value to the current attribute which is lower or equal to the current threshold. It is required to compute the multi-label confusion matrices for all labels under the current condition derived from the current attribute value as a threshold.

The only labels which are predicted by a rule are the minority labels, i.e., the least frequent value in the training data set. This is due to the fact that predicting the opposite of the minority labels would be equivalent to abstaining from a prediction or predicting the default rule. This corresponds to *rule-dependent* evaluation (Rapp, 2016). Another possible approach to handling unpredicted labels would be treating unpredicted labels as predicted to be irrelevant during the evaluation. This *rule-independent* evaluation strategy however has lead to worse performance, increased runtime and longer learned decision lists in previous experiments (cf. Rapp (2016)) and therefore it is not used in this work.

Note that while the matrices  $C_{covered}$  and  $C_{default}$  contain four elements according to the classic notation of confusion matrices, the evaluation function  $\delta$  is the adapted version of a multi-label evaluation metric which requires a confusion matrix with eight different entries. This is due to the fact that  $C_{covered}$  and  $C_{default}$  do not respect the dimension of coverage (see chapter 2.3.2) as they already imply whether a label is covered or not. The operator `++` is the increment operator, which increments the value of the preceding operand by one.

---

**Algorithm 7.** Algorithm UPDATESEARCH for aggregating the confusion matrix  $C_{covered}$ 

---

**Input:** Training data instance  $(X, Y)$ , matrix  $C_{covered}$

```
1: procedure UPDATESEARCH
2:   Let  $\hat{y}_1 \dots \hat{y}_{|\mathbb{L}|}$  be the minority labels as a prediction
3:   for  $\lambda_i \in \mathbb{L}$  do
4:     if ISUNCOVERED( $y_i$ ) then
5:       if  $\hat{y}_i = 1$  then
6:         if  $y_i = 1$  then
7:            $C_{covered}^{TP}++$ 
8:         else
9:            $C_{covered}^{FP}++$ 
10:        end if
11:       else
12:        if  $y_i = 1$  then
13:           $C_{covered}^{FN}++$ 
14:        else
15:           $C_{covered}^{TN}++$ 
16:        end if
17:       end if
18:     end if
19:   end for
20: end procedure
```

---

Algorithm ISUNCOVERED returns whether or not the label value  $y_i$  is already covered either by the default rule or an other rule induced before. This is implemented using a matrix mask, which is updated every time a new rule is induced. However, as the implementation and usage of this matrix is not important for the implementation of the algorithm proposed in this work, it is not shown explicitly.

Algorithm 8 RESETSEARCH resets the aggregated confusion matrix  $C_{covered}$  by assigning the value 0 to each entry of the confusion matrix. This required for nominal attributes as well as at the beginning of each new condition search.

---

**Algorithm 8.** Algorithm RESETSEARCH for resetting the state of the condition search

---

**Input:** Matrix  $C_{covered}$

```
1: procedure RESETSEARCH
2:    $C_{covered}^{TP} = 0$ 
3:    $C_{covered}^{FP} = 0$ 
4:    $C_{covered}^{FN} = 0$ 
5:    $C_{covered}^{TN} = 0$ 
6: end procedure
```

---

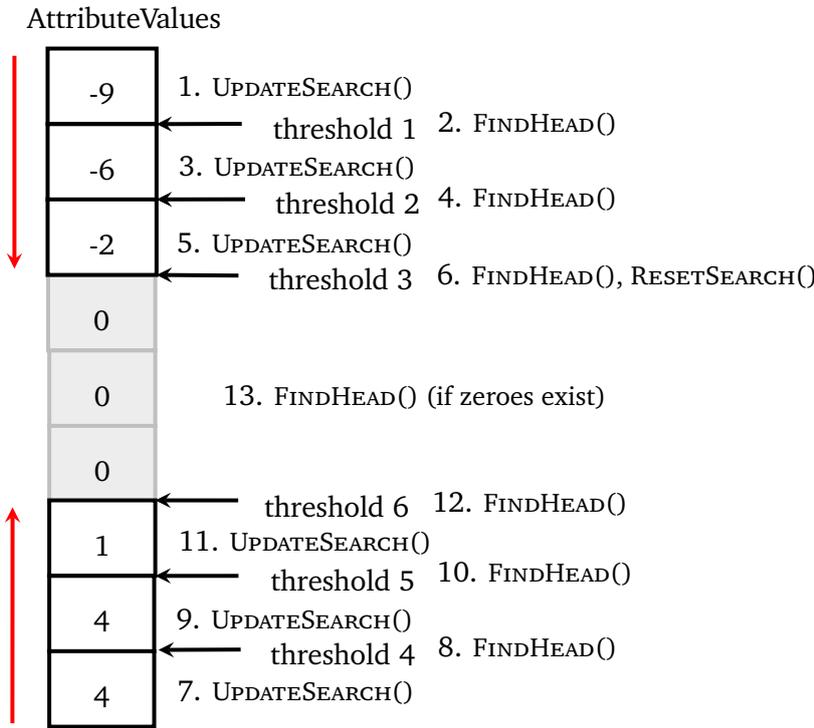
### 3.2.2. Sparse Condition Finding

Sparse matrices are matrices in which most elements are zero. In machine learning, sparse representations of matrices often do not explicitly contain zero values. As sparse matrix representations are common in

the area of machine learning, this chapter briefly introduces how the sparsity of matrices were exploited in the experiments on the implemented algorithm. Sparse condition finding is not part of this work, as it is implemented in the BOOMER framework. Its idea is based on "sparsity-aware split finding" (see Chen and Guestrin (2016)).

Due to the sorting of attribute values, sparse matrices can be handled efficiently with a few changes to the original algorithm. Instead of only iterating through the array of attribute values once, the algorithm processes the attribute values once from the top (negative values) and once from the bottom up (positive values) to the value 0 in the sorted array. After that, if zeroes exist in the sparse matrix, the best head is searched again with respect to the zeroes, which are not explicitly featured in the sparse matrix.

Figure 6.: Sparse Condition Finding



Note that the state of the condition search is reset after all negative attribute values have been processed. This is necessary, so that the confusion matrix for uncovered examples ( $C_{default} - C_{covered}$ ) can still be calculated correctly when the positive attribute values are processed.

### 3.3. Head Refinement

Algorithm 9 FINDHEAD is called by algorithm 6 REFINERULE and finds the best head for a given rule according to a decomposable evaluation function  $\delta$ . In the experiments, two different approaches for head refinement were used. Single-label head refinement corresponds to learning only single-label head rules, while multi-label head refinement is also able to induce multi-label head rules. In order to increase the efficiency of the implemented algorithm, the maximum number of head refinements per rule was restricted to one in the

conducted experiments. This means that rules keep the first head assigned to them after the first condition refinement. Testing other restrictions or unlimited head refinements is left for future work. The following two chapters describe both approaches, which can be used interchangeably.

---

**Algorithm 9.** Algorithm FINDHEAD for finding and setting the best head for a rule

---

**Input:** Rule  $r$ , relaxation lift function  $\rho$  with maximum value  $\rho_{max}$ , whether or not uncovered examples shall be evaluated  $evaluateUncovered$

**Output:** The rule with the best head  $r_{best}$

```

1: procedure FINDMULTILABELHEAD
2:   if Multi-label head refinement shall be used: then
3:     return FINDMULTILABELHEAD( $R, \rho, evaluateUncovered$ )
4:   else
5:     return FINDSINGLELABELHEAD( $R, evaluateUncovered$ )
6:   end if
7: end procedure

```

---

### 3.3.1. Single-Label Head Refinement

Single-label head refinement is used as a baseline approach for comparison with the multi-label head refinement approach. Algorithm 10 FINDSINGLELABELHEAD simply finds the label for which the evaluation score according to evaluation function  $\delta$  is the highest and returns the rule predicting this label.

---

**Algorithm 10.** Algorithm FINDSINGLELABELHEAD for finding and setting the best single-label head for a rule

---

**Input:** Rule  $r$ , whether or not uncovered examples shall be evaluated  $evaluateUncovered$

**Output:** The rule with the best head  $r_{best}$

```

1: procedure FINDSINGLELABELHEAD
2:    $r_{best} = r$ 
3:    $scores = EVALUATELABELPREDICITONS(evaluateUncovered)$ 
4:   for all  $h_i \in scores$  do
5:      $r = r.body \rightarrow \lambda_i$ 
6:      $r.h = h_i$ 
7:     if  $r.h > r_{best}.h$  then
8:        $r_{best} = r$ 
9:     end if
10:  end for
11:  return  $r_{best}$ 
12: end procedure

```

---

### 3.3.2. Multi-Label Head Refinement

As rules with multiple labels in the head can be learned with multi-label head refinement, the most promising label combinations need to be evaluated. First, the scores obtained by algorithm 12 EVALUATELABELPREDICITONS are sorted in descending order. The subroutine SORT returns a list of tuples  $(\lambda_i, h_i)$  which corresponds

to the sorted labels and their evaluation scores determined in Algorithm 12 EVALUATELABELPREDICTIONS respectively.

Then, instead of performing an exhaustive search through all possible label-combinations, only the labels with the highest evaluation scores are successively considered until the current best evaluation score cannot be reached anymore by adding more labels according to relaxed pruning based on decomposability. In order to obtain a single score for a rule, based on the evaluation scores of the labels in the rule head, the label-wise averaged total score is multiplied with the lift provided by the relaxation lift function  $\rho$ . The head of the rule which reached the highest lifted score is then returned as the best head.

---

**Algorithm 11.** Algorithm FINDMULTILABELHEAD for finding and setting the best multi-label head for a rule

---

**Input:** Rule  $r$ , relaxation lift function  $\rho$  with maximum value  $\rho_{max}$ , whether or not uncovered examples shall be evaluated *evaluateUncovered*

**Output:** The rule with the best head  $r_{best}$

```

1: procedure FINDMULTILABELHEAD
2:    $r_{best} = r$ 
3:    $h_{total} = 0$ 
4:    $scores = \text{EVALUATELABELPREDICITONS}(\text{evaluateUncovered})$ 
5:    $scores_{sorted} = \text{SORT}(scores)$ 
6:   for all  $(\lambda_i, h_i) \in scores_{sorted}$  do
7:      $r = r.body \rightarrow r.head \cup \lambda_i$ 
8:      $h_{total} = h_{total} + h_i$ 
9:      $h = \frac{h_{total}}{|r.head|}$ 
10:     $r.h = h * \rho(|r.head|)$  ▷ Apply relaxation lift
11:    if  $r.h \geq r_{best}.h$  then
12:       $r_{best} = r$ 
13:    else if  $\rho_{max} * h < r_{best}.h$  then ▷ Relaxed pruning according to decomposability
14:      break
15:    end if
16:  end for
17:  return  $r_{best}$ 
18: end procedure

```

---

### 3.4. Measuring Performance

---

Algorithm 12 EVALUATELABELPREDICTIONS calculates the evaluation scores for rules predicting the minority labels for either all covered examples or all uncovered examples. In order to obtain the multi-label confusion matrix for all labels under the current condition,  $C_{covered}$  has to be concatenated with the result of the element-wise subtraction  $C_{default} - C_{covered}$ . The multi-label confusion matrix for the inverted condition can also be obtained by swapping the confusion matrix for all covered examples  $C_{covered}$  with the confusion matrix for all uncovered examples  $C_{default} - C_{covered}$ . Note that the union operator  $\cup$  denotes the concatenation of two classic confusion matrices with four entries to one multi-label confusion matrix with eight entries in algorithm 12. The first operand contains all elements which will be considered covered in the resulting multi-label confusion matrix, while the second operand contains all elements which will be considered uncovered.

---

**Algorithm 12.** Algorithm EVALUATELABELPREDICTIONS for computing the evaluation scores for all labels based on matrices  $C_{covered}$  and  $C_{default}$

---

**Input:** Evaluation function  $\delta$ , whether or not uncovered examples shall be evaluated

**Output:** Computed evaluation scores for all labels  $scores$

```
1: procedure EVALUATELABELPREDICTIONS
2:    $scores = (0)^{|\mathbb{L}|}$  ▷ Initialize  $scores$  as an array of length  $|\mathbb{L}|$ 
3:   for  $\lambda_i \in \mathbb{L}$  do
4:     if confusion matrix for uncovered examples shall be evaluated then
5:        $C = (C_{default} - C_{covered}) \cup C_{covered}$ 
6:     else
7:        $C = C_{covered} \cup (C_{default} - C_{covered})$ 
8:     end if
9:      $scores[i] = \delta(C)$ 
10:  end for
11:  return  $scores$ 
12: end procedure
```

---

---

## 4. Evaluation of Results

---

In this chapter, the best configurations of selected hyperparameters for different data sets are determined in order to measure their influence on the performance of the multi-label rule learner. First, the experiments performed are described, including definitions of the used evaluation method, tested data sets, tuned parameters and specific implementation settings including the selected heuristic and relaxation lift function. After that, the results of the experiments as well as an evaluation of the different models generated are presented.

---

### 4.1. Experimental Setup

---

This chapter describes the experimental setup which was used to find the optimal hyperparameter configurations and on which data sets they were tested. It also includes which hyperparameters were tuned and why they were chosen over other parameters.

#### 4.1.1. Bootstrap Bias Corrected Cross Validation

Traditionally, nested cross validation is often used in order to find the optimal hyper-parameter configuration (cf. Varma and Simon (2006)). However, according to Tsamardinos et al. (2017), nested cross validation can be replaced by bootstrapping the selection of the optimal configuration on out-of-sample predictions of each configuration for tuning hyper-parameters, yielding better performance and a smaller variance and bias. In this work, Bootstrap Bias Corrected Cross Validation (BBC-CV) is used with 50 bootstrapping iterations and the target evaluation measure Hamming loss. Chapter 4 contains a definition of Hamming loss and other measures used to evaluate the performance of the tested configurations.

#### 4.1.2. Data Sets

The implemented rule learner was tested with five different data sets taken from the Mulan project (Tsoumakas et al., 2011). Table 4 lists relevant attributes of the used data sets. These are namely the name of the data set, the domain of the input examples, the number of examples, the number of nominal attributes, the number of numeric attributes, the total number of unique labels, the average number of labels per example called cardinality, the average percentage of relevant labels called density and the number of distinct label combinations in the data set.

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
birds	Audio	645	2	258	19	1.014	0.053	133
yeast	Biology	2417	0	103	14	4.237	0.303	198
medical	Text	978	1449	0	45	1.245	0.028	94

Table 4.: Relevant characteristics of the data sets used in the experiments

### 4.1.3. Tuned Parameters

Table 5 shows which parameters were tuned and which values of these parameters were tested in order to find the optimal configuration for the chosen data sets.

- The parameter **head refinement** describes the head refinement strategy which was used to learn rule heads, with the value *single-label* corresponding to learning only single-label head rules and the value *multi-label* referring to learning multi-label head rules as well.
- The parameter **max\_rules** controls the maximum number of rules learned by the algorithm. If no other stopping criterion is specified, the algorithm stops as soon as all examples are covered by default. However, this would lead to overfitting as mentioned in chapter 2.2.2. As stopping criteria are left for future work, they are not considered in the proposed algorithm. Instead, the **max\_rules** parameter is handled as a tuned parameter. Models with less than the maximal value for **max\_rules** are obtained by cutting off rules from the end of the decision list of the model until only the default rule is left, which corresponds to the **max\_rules** parameter 1.
- For the evaluation function M-estimate, the only parameter **m** is tested with the values 0 and all powers of 2 between 1 and 512 inclusively.
- When Relaxed Pruning is tested, only the parameter **max\_lift** is tuned, while the parameters **curvature** and **peak\_label** stay fixed at 1 and the rounded up cardinality of the data set respectively.

Parameter	Values
<i>Base Learner</i>	
head refinement	single-label, multi-label
max_rules	[1,∞)
<i>M-Estimate</i>	
m	0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512
<i>Relaxed Pruning (only with multi-label head refinement)</i>	
max_lift	1, 1.02, 1.04, 1.08, 1.16, 1.32
curvature	1
peak_label	[Cardinality]

Table 5.: Tuned parameters and tested values. The **peak\_label** parameter is chosen as the cardinality, rounded up to the next positive integer

---

As shown in chapter 2.3.3, many heuristics can possibly be used in order to evaluate the quality of a multi-label head rule. However, according to Rapp, Loza Mencía, et al. (2019), a heuristic which is able to trade off between consistency and coverage, like the F-measure or M-estimate heuristic, may greatly improve performance, if they are fine-tuned depending on the used data sets. As listed in table 5, only the M-estimate heuristic with the tuned parameter  $m$  was used in this implementation.

---

## 4.2. Predictive Quality

---

In this chapter, the experimental results in terms of predictive quality are presented. First, the metrics used to measure the performance of models will be described. After that, an analysis of the obtained results is conducted. All data evaluated can be found in appendix A.

- **Hamming Accuracy:**  
The Hamming accuracy metric corresponds to the percentage of correctly classified labels and is defined as in equation 2.4.
- **Example-based Precision:**  
The precision metric corresponds to the fraction of correctly classified labels among all predicted labels and is defined as in Equation 2.2. For this metric, the example-wise averaging strategy is used as defined in equation 2.13.
- **Example-based Recall:**  
The recall metric corresponds to the fraction of predicted labels among all relevant labels and is defined as in equation 2.6. For this metric, the example-wise averaging strategy is used as defined in equation 2.13.
- **Example-based F1-Measure:**  
The F1-measure computes the harmonic mean between precision and recall and is defined as in equation 2.9. For this metric, the example-wise averaging strategy is used as defined in equation 2.13.
- **Micro-averaged Precision:**  
The precision metric corresponds to the fraction of correctly classified labels among all predicted labels and is defined as in Equation 2.2. For this metric, the micro-averaging was used as defined in equation 2.12.
- **Micro-averaged Recall:**  
The recall metric corresponds to the fraction of predicted labels among all relevant labels and is defined as in equation 2.6. For this metric, the micro-averaging was used as defined in equation 2.12.
- **Micro-averaged F1-Measure:**  
The F1-measure computes the harmonic mean between precision and recall and is defined as in equation 2.9. For this metric, the micro-averaging was used as defined in equation 2.12.

Table 6 shows the performance of the best configurations according to the hamming accuracy metric for all data sets using either single-label head refinement or multi-label head refinement. In almost all cases, the approach using multi-label head refinement outperforms the approach using single-label head refinement.

data set	head ref.	Ha.Acc.	Ex.Prec.	Ex.Rec.	Ex.F1.	Mi.Prec.	Mi.Rec.	Mi.F1
birds	single-label	94.85%	99.14%	48.85%	49.38%	81.82%	3.83%	7.32%
	multi-label	94.78%	98.28%	49.17%	49.50%	66.67%	3.40%	6.48%
emotions	single-label	70.85%	97.91%	5.12%	6.14%	77.78%	5.37%	10.05%
	multi-label	71.71%	95.58%	9.38%	11.24%	76.00%	9.72%	17.23%
yeast	single-label	77.09%	74.91%	34.92%	46.74%	74.97%	36.84%	49.40%
	multi-label	77.45%	75.18%	35.68%	47.27%	76.01%	37.61%	50.32%
medical	single-label	97.38%	99.27%	7.39%	7.54%	89.86%	6.77%	12.59%
	multi-label	97.54%	99.59%	12.68%	13.45%	97.37%	12.12%	21.55%
scene	single-label	82.80%	99.89%	4.27%	4.29%	97.44%	4.03%	7.73%
	multi-label	83.94%	98.75%	11.66%	11.83%	90.08%	11.55%	20.47%

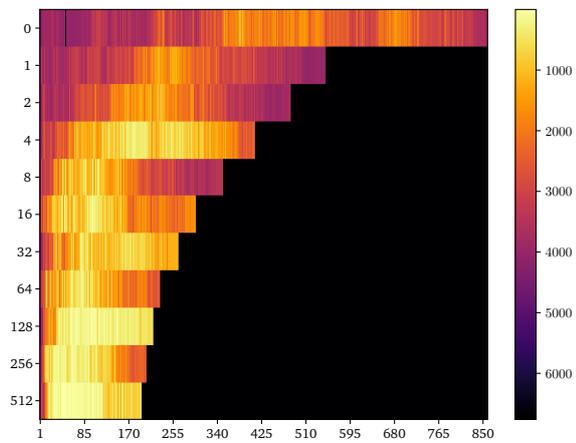
Table 6.: Best performances all models learned.

#### 4.2.1. Single-Label Head Refinement

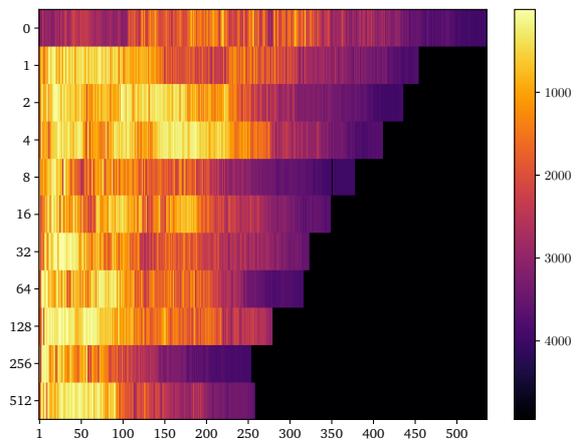
The heat maps in Figure 7a, 7b, 7c, 7d and 7e depict the average ranks of all tested configurations using the single-label head refinement. The  $m$  value parameter of the M-estimate heuristic on the y-axis is plotted against the  $\text{max\_rules}$  parameter on the x-axis. For all data sets, the number of rules learned until all examples are covered decreases when a higher value for  $m$  is chosen. The only exception is an increase of learned rules when the  $m$  value is increased from 256 to 512 on the data set birds. The performance of the learned model also tends to increase for many data sets with a rising  $m$  value up to a certain point dependent on the data set. The value 0 seems to be a bad choice for the parameter  $m$  in all cases. This is most likely due to the fact that the M-estimate heuristic with the parameter value 0 for  $m$  is equivalent to the precision metric and therefore ignores the coverage of rules learned. This results in a bad performance and a large number of rules which have to be learned until all examples are covered. Especially for data sets with many instances and labels, relatively many rules have to be learned until all examples are covered. While removing rules from the end of the decision list in order to prevent overfitting as described in chapter 4.1.3 is effective for the data sets scene, birds, emotions and yeast, the optimal number of rules learned is almost equal to the maximum number of rules learned for the data set medical in most cases.

#### 4.2.2. Multi-Label Head Refinement

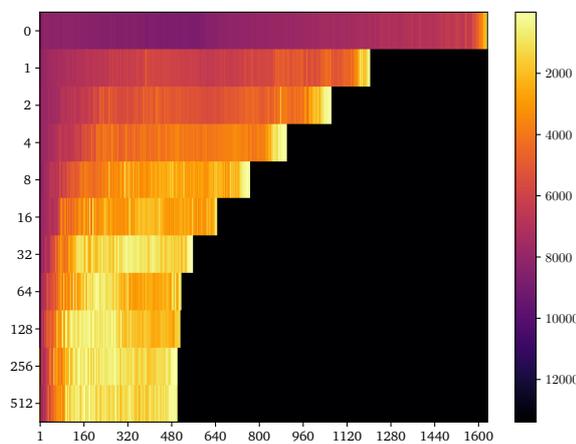
The heat maps in Figure 8a, 8b, 8c, 8d and 8e depict the average ranks of all tested configurations using the multi-label head refinement. The  $\text{max\_lift}$  value for relaxed pruning on the y-axis is plotted against the  $m$  value parameter for the M-estimate heuristic on the x-axis. The performance-wise advantage of a high  $m$  value is also reflected in the results for the multi-label head refinement experiments. For most data sets, a global maximum for the  $m$  value is visible, even though some local maxima are visible for the data sets birds and arguably for other data sets. Given an  $m$  value of 64 or 128, a high value for the parameter  $\text{max\_lift}$  seems to be beneficial for the data set yeast, which has a high cardinality and consequently a high  $\text{peak\_label}$  value. The parameters  $m$  and  $\text{max\_lift}$  do not seem to be independent of each other for all other data sets



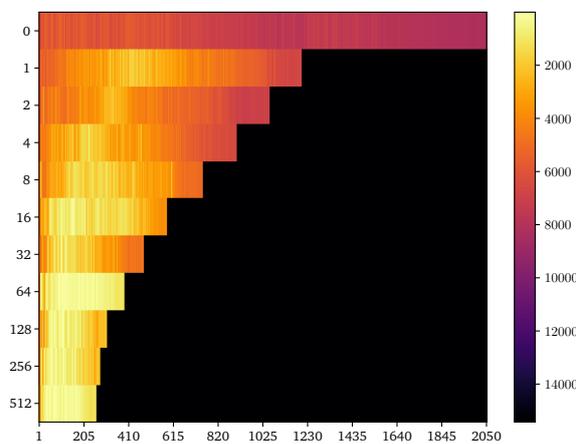
(a) data set emotions



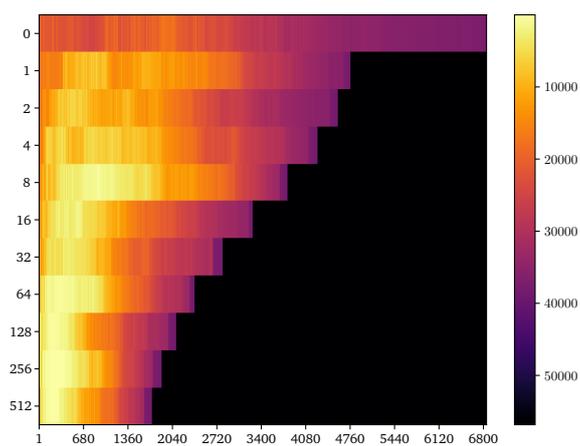
(b) data set birds



(c) data set medical



(d) data set scene



(e) data set yeast

Figure 7.: Average ranks of all tested configurations using the single-label head refinement

---

as well. This may be due to the fact that they both influence the coverage of learned rules in different ways. Note that some results may be influenced by the fact that for multi-label head refinement, not all possible values for the parameter `max_rules` could be tested due to resource limitations. Only every value divisible by 5, up to the maximum number of rules, was tested for the parameter `max_rules`.

---

### 4.3. Model Characteristics

---

In order to compare the quality of the different learned models, not only the predictive performance but also selected other characteristics are statistically evaluated. The following model characteristics have been obtained in the conducted experiments:

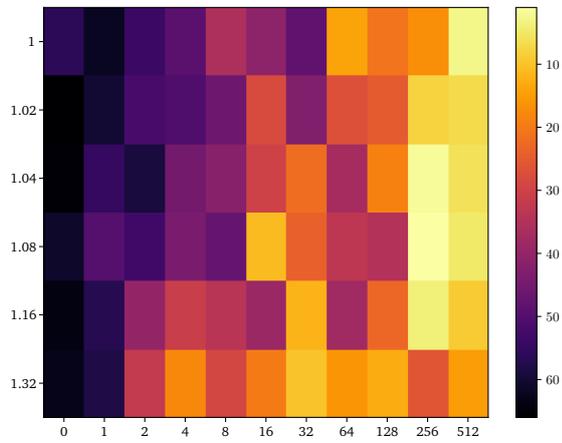
- **#rules** describes the number of rules learned until all examples have been covered.
- **Avg.Con.** corresponds to the average number of conditions per learned rule.
- **opt. #rules** corresponds the number of rules for which the model reached the highest performance according to the Hamming accuracy metric.
- **runtime** describes the time required until the model was learned in the format HH:MM:SS. The experiments were conducted on the computation cluster of the Knowledge Engineering Group at TU Darmstadt.

The following characteristics are only valid if the multi-label head refinement was used:

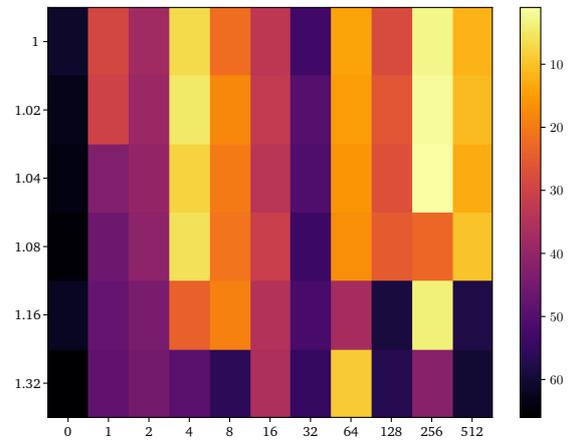
- **%MLHR** describes the percentage of multi-label head rules learned among all rules of the decision list, excluding the default rule.
- **Avg.MLH** corresponds to the average number of labels in a multi-label head rule among all multi-label head rules.

Tables 7, 8, 9, 10 and 11 show the model characteristics of all models learned with the multi-label head refinement, while tables 12, 13, 14 and 15, 16 show the model characteristics of all models learned with the single-label head refinement. The parameters **head ref.**, **m** and **data set** describe which parameters were used when the models were learned:

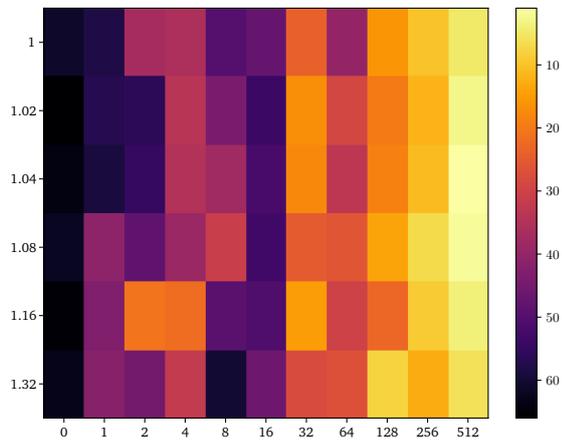
- The parameter **head ref.** describes the head refinement strategy which was used to learn rule heads, with the value *single-label* corresponding to learning only single-label head rules and the value *multi-label* referring to learning multi-label head rules as well.
- The parameter **m** corresponds to the m value used in the M-estimate heuristic
- The **data set** is the data set on which the model was learned.
- The parameter **l** is only valid for models using the multi-label head refinement and is the short notation for the maximum lift used as a parameter for the peak relaxation lift function.



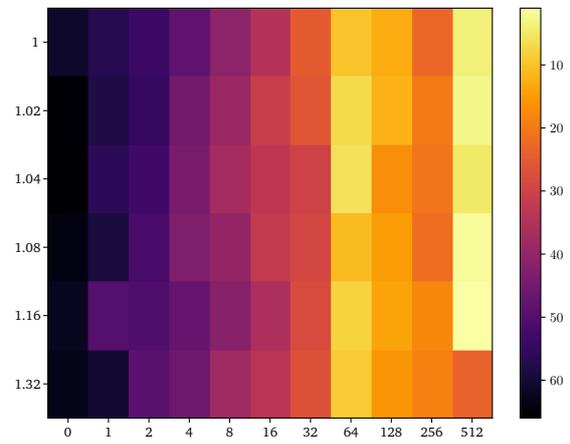
(a) data set emotions



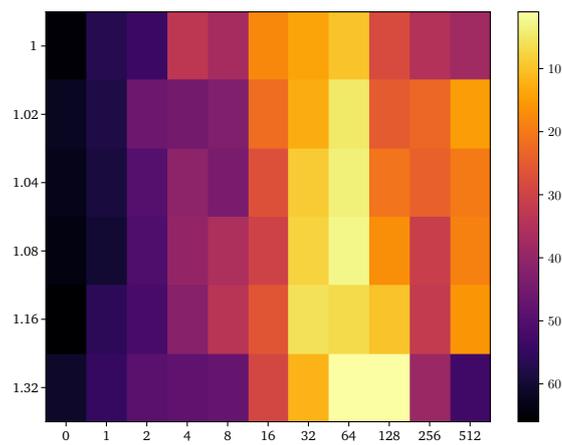
(b) data set birds



(c) data set medical



(d) data set scene



(e) data set yeast

Figure 8.: Average ranks of all tested configurations using the multi-label head refinement

---

## 5. Conclusion

---

This final chapter summarizes the most important results of this work and gives an overview over possible future improvements of the implemented algorithm.

---

### 5.1. Summary

---

In this work, an algorithm based on the separate-and-conquer algorithm proposed by Rapp (2016), combining multiple approaches suggested in recent studies (cf. Klein (2018); Hefter (2020); Loza Mencía (2012)), was implemented and empirically tested. The algorithm was implemented in the BOOMER framework (Rapp, Mencía, et al., 2020) and contains algorithmic improvements over a naive implementation of the base algorithm. The test results of different approaches and hyperparameter settings show the performance of the suggested algorithm under selected configurations. As the examinations showed, the learned number of rules can be reduced by increasing the value of the parameter  $m$  of the M-estimate heuristic, without trading off much performance. In fact, a  $m$  value between 64 and 512 lead to the best possible results in many cases, while producing relatively short rule lists. The evaluation shows that the efficiency and predictive performance of the implemented algorithm is comparable with other multi-label classification algorithms.

---

### 5.2. Future Work

---

The M-estimate metric was the only metric used in the experiments conducted in this work. Other metrics, for example the F-measure metric, can be used with the implemented algorithm as well as long as the label-wise averaging strategy is used or they otherwise fulfill the conditions required for decomposability. Furthermore, the implemented algorithm could be improved by using a pruning mechanism like IREP (see Johannes and Widmer (1996)).

As only trivial stopping criteria are used in the implemented algorithm, a single or multiple stopping criteria, for example based on the coverage of the training data set, are required.

Another improvable aspect of the implemented algorithm is the parameterized maximum number of head refinements mentioned in chapter 3.3. As its value was 1 during all experiments conducted in this work, other values, as well as other approaches to restricting the number of head refinements, may improve the quality of learned models.

Label conditions are conditions which depend on labels predicted by previous rules in the decision list. Rules may partially or fully depend on label conditions and therefore, they can exploit or expose label dependencies. In this work, label conditions were not considered, as label dependencies can also be expressed with multi-label rule heads. However, the induction of label conditions may yield improvements to the proposed algorithm.

## A. Performance Evaluations

Table 7.: Performance of multi-label models learned from the data set birds

data set	parameters			performance measures						
	head ref.	m	l	Ha.Acc.	Ex.Prec.	Ex.Rec.	Ex.F1.	Mi.Prec.	Mi.Rec.	Mi.F1
birds	multi-label	0	-	92.35%	69.74%	61.88%	55.12%	32.43%	35.82%	34.04%
birds	multi-label	0	1.02	92.76%	76.09%	59.79%	52.53%	33.33%	31.34%	32.31%
birds	multi-label	0	1.04	92.68%	74.53%	59.79%	50.97%	32.81%	31.34%	32.06%
birds	multi-label	0	1.08	92.60%	72.97%	59.79%	50.97%	32.31%	31.34%	31.82%
birds	multi-label	0	1.16	92.60%	72.97%	59.79%	50.97%	32.31%	31.34%	31.82%
birds	multi-label	0	1.32	92.85%	72.79%	60.57%	51.64%	34.85%	34.33%	34.59%
birds	multi-label	1	-	93.50%	73.75%	64.22%	57.77%	41.18%	41.79%	41.48%
birds	multi-label	1	1.02	93.17%	73.10%	65.78%	58.76%	39.19%	43.28%	41.13%
birds	multi-label	1	1.04	93.09%	71.54%	65.78%	57.20%	38.67%	43.28%	40.85%
birds	multi-label	1	1.08	93.09%	71.41%	65.26%	56.94%	38.36%	41.79%	40.00%
birds	multi-label	1	1.16	93.17%	69.68%	66.33%	57.28%	39.19%	43.28%	41.13%
birds	multi-label	1	1.32	92.68%	67.60%	64.17%	55.79%	35.90%	41.79%	38.62%
birds	multi-label	2	-	93.91%	72.58%	67.08%	57.59%	44.93%	46.27%	45.59%
birds	multi-label	2	1.02	93.42%	69.27%	67.08%	56.93%	41.33%	46.27%	43.66%
birds	multi-label	2	1.04	93.34%	67.71%	67.08%	55.37%	40.79%	46.27%	43.36%
birds	multi-label	2	1.08	93.42%	70.62%	67.08%	56.80%	41.33%	46.27%	43.66%
birds	multi-label	2	1.16	93.50%	69.55%	67.73%	57.32%	42.11%	47.76%	44.76%
birds	multi-label	2	1.32	93.09%	64.61%	62.01%	54.20%	37.31%	37.31%	37.31%
birds	multi-label	4	-	94.16%	71.93%	66.56%	57.30%	46.97%	46.27%	46.62%
birds	multi-label	4	1.02	94.16%	71.56%	66.56%	57.76%	46.97%	46.27%	46.62%
birds	multi-label	4	1.04	93.67%	70.39%	66.56%	55.68%	43.06%	46.27%	44.60%
birds	multi-label	4	1.08	93.50%	67.27%	66.56%	55.68%	41.89%	46.27%	43.97%
birds	multi-label	4	1.16	93.67%	68.83%	67.21%	54.97%	43.42%	49.25%	46.15%
birds	multi-label	4	1.32	92.43%	69.45%	66.20%	56.27%	34.57%	41.79%	37.84%
birds	multi-label	8	-	94.33%	74.90%	65.91%	57.70%	48.48%	47.76%	48.12%
birds	multi-label	8	1.02	93.91%	72.45%	65.39%	57.31%	44.93%	46.27%	45.59%
birds	multi-label	8	1.04	93.42%	66.56%	65.39%	55.22%	41.33%	46.27%	43.66%
birds	multi-label	8	1.08	93.50%	68.91%	69.04%	57.28%	42.50%	50.75%	46.26%
birds	multi-label	8	1.16	93.50%	75.29%	66.35%	59.25%	42.31%	49.25%	45.52%
birds	multi-label	8	1.32	92.02%	70.10%	65.96%	56.08%	34.04%	47.76%	39.75%
birds	multi-label	16	-	94.90%	78.67%	65.60%	61.59%	54.10%	49.25%	51.56%
birds	multi-label	16	1.02	94.08%	71.85%	64.24%	55.72%	46.15%	44.78%	45.45%
birds	multi-label	16	1.04	93.83%	68.65%	65.03%	55.92%	44.12%	44.78%	44.44%
birds	multi-label	16	1.08	94.00%	71.22%	66.46%	58.99%	45.71%	47.76%	46.72%

birds	multi-label	16	1.16	92.76%	68.68%	65.00%	56.85%	37.04%	44.78%	40.54%
birds	multi-label	16	1.32	94.08%	75.52%	67.03%	59.50%	46.15%	44.78%	45.45%
birds	multi-label	32	-	94.90%	75.55%	65.21%	57.48%	54.10%	49.25%	51.56%
birds	multi-label	32	1.02	94.82%	73.85%	65.21%	57.24%	53.23%	49.25%	51.16%
birds	multi-label	32	1.04	94.24%	69.04%	66.77%	56.18%	47.89%	50.75%	49.28%
birds	multi-label	32	1.08	93.42%	64.92%	64.61%	51.53%	41.77%	49.25%	45.21%
birds	multi-label	32	1.16	93.34%	70.05%	67.01%	55.94%	41.67%	52.24%	46.36%
birds	multi-label	32	1.32	94.65%	75.91%	68.07%	61.30%	51.56%	49.25%	50.38%
birds	multi-label	64	-	94.65%	72.14%	67.81%	58.14%	51.61%	47.76%	49.61%
birds	multi-label	64	1.02	94.24%	67.32%	68.07%	55.01%	47.83%	49.25%	48.53%
birds	multi-label	64	1.04	93.42%	68.75%	64.69%	51.06%	41.10%	44.78%	42.86%
birds	multi-label	64	1.08	92.93%	68.26%	65.08%	52.37%	36.99%	40.30%	38.57%
birds	multi-label	64	1.16	92.93%	73.46%	58.83%	49.58%	36.62%	38.81%	37.68%
birds	multi-label	64	1.32	93.17%	65.94%	68.07%	49.76%	39.47%	44.78%	41.96%
birds	multi-label	128	-	94.08%	63.65%	67.55%	54.72%	46.48%	49.25%	47.83%
birds	multi-label	128	1.02	94.24%	73.26%	69.06%	56.43%	47.76%	47.76%	47.76%
birds	multi-label	128	1.04	94.08%	70.80%	68.52%	58.35%	46.75%	53.73%	50.00%
birds	multi-label	128	1.08	94.49%	76.32%	69.37%	59.29%	50.00%	53.73%	51.80%
birds	multi-label	128	1.16	93.01%	65.29%	63.36%	47.71%	37.84%	41.79%	39.72%
birds	multi-label	128	1.32	93.01%	74.34%	59.90%	50.02%	35.94%	34.33%	35.11%
birds	multi-label	256	-	94.33%	71.64%	67.55%	58.29%	48.48%	47.76%	48.12%
birds	multi-label	256	1.02	94.33%	70.21%	68.59%	57.62%	48.44%	46.27%	47.33%
birds	multi-label	256	1.04	92.35%	62.96%	66.98%	51.74%	34.52%	43.28%	38.41%
birds	multi-label	256	1.08	93.91%	73.41%	64.90%	53.68%	44.62%	43.28%	43.94%
birds	multi-label	256	1.16	92.85%	69.90%	64.48%	54.19%	37.18%	43.28%	40.00%
birds	multi-label	256	1.32	91.37%	63.88%	66.43%	51.38%	31.37%	47.76%	37.87%
birds	multi-label	512	-	94.24%	71.15%	68.78%	58.66%	47.95%	52.24%	50.00%
birds	multi-label	512	1.02	93.91%	71.12%	66.51%	56.02%	45.07%	47.76%	46.38%
birds	multi-label	512	1.04	94.82%	76.95%	70.21%	65.82%	53.12%	50.75%	51.91%
birds	multi-label	512	1.08	93.83%	71.35%	69.51%	55.22%	44.44%	47.76%	46.04%
birds	multi-label	512	1.16	93.17%	70.36%	71.54%	60.64%	39.74%	46.27%	42.76%
birds	multi-label	512	1.32	93.59%	71.66%	65.13%	54.13%	42.47%	46.27%	44.29%

Table 8.: Performance of multi-label models learned from the data set emotions

parameters				performance measures						
data set	head ref.	m	l	Ha.Acc.	Ex.Prec.	Ex.Rec.	Ex.F1.	Mi.Prec.	Mi.Rec.	Mi.F1
emotions	multi-label	0	-	66.95%	64.52%	50.28%	39.98%	46.90%	48.18%	47.53%
emotions	multi-label	0	1.02	69.77%	78.22%	44.63%	36.32%	51.58%	44.55%	47.80%
emotions	multi-label	0	1.04	69.77%	78.22%	44.63%	36.32%	51.58%	44.55%	47.80%
emotions	multi-label	0	1.08	69.49%	78.22%	43.79%	35.19%	51.06%	43.64%	47.06%
emotions	multi-label	0	1.16	68.08%	73.14%	44.35%	35.19%	48.45%	42.73%	45.41%
emotions	multi-label	0	1.32	67.23%	67.66%	49.44%	40.19%	47.37%	49.09%	48.21%
emotions	multi-label	1	-	68.93%	66.07%	48.87%	42.77%	50.00%	48.18%	49.07%
emotions	multi-label	1	1.02	69.49%	68.19%	50.00%	42.95%	50.96%	48.18%	49.53%
emotions	multi-label	1	1.04	70.06%	74.04%	46.61%	40.52%	52.08%	45.45%	48.54%
emotions	multi-label	1	1.08	68.64%	63.95%	44.92%	38.38%	49.48%	43.64%	46.38%

emotions	multi-label	1	1.16	70.62%	72.71%	43.79%	39.81%	53.49%	41.82%	46.94%
emotions	multi-label	1	1.32	71.47%	71.38%	49.44%	41.95%	54.46%	50.00%	52.13%
emotions	multi-label	2	-	70.06%	69.75%	54.24%	44.87%	51.75%	53.64%	52.68%
emotions	multi-label	2	1.02	70.34%	65.88%	52.54%	45.52%	52.25%	52.73%	52.49%
emotions	multi-label	2	1.04	69.49%	67.18%	49.44%	43.14%	50.96%	48.18%	49.53%
emotions	multi-label	2	1.08	68.93%	62.49%	48.02%	41.34%	50.00%	48.18%	49.07%
emotions	multi-label	2	1.16	71.19%	69.77%	49.44%	43.71%	54.00%	49.09%	51.43%
emotions	multi-label	2	1.32	70.06%	64.41%	46.05%	40.38%	52.13%	44.55%	48.04%
emotions	multi-label	4	-	71.19%	70.14%	55.08%	47.03%	53.77%	51.82%	52.78%
emotions	multi-label	4	1.02	70.34%	64.10%	54.24%	44.99%	52.25%	52.73%	52.49%
emotions	multi-label	4	1.04	70.06%	67.77%	50.28%	42.78%	51.89%	50.00%	50.93%
emotions	multi-label	4	1.08	67.51%	57.74%	43.22%	38.37%	47.52%	43.64%	45.50%
emotions	multi-label	4	1.16	71.75%	70.00%	48.59%	44.32%	55.32%	47.27%	50.98%
emotions	multi-label	4	1.32	71.47%	70.20%	51.98%	44.56%	54.55%	49.09%	51.67%
emotions	multi-label	8	-	73.45%	71.38%	53.11%	50.54%	58.00%	52.73%	55.24%
emotions	multi-label	8	1.02	74.58%	75.28%	57.06%	54.13%	59.62%	56.36%	57.94%
emotions	multi-label	8	1.04	74.01%	72.03%	56.50%	53.28%	58.65%	55.45%	57.01%
emotions	multi-label	8	1.08	70.34%	66.44%	50.85%	44.07%	52.48%	48.18%	50.24%
emotions	multi-label	8	1.16	75.42%	71.75%	49.72%	47.34%	63.22%	50.00%	55.84%
emotions	multi-label	8	1.32	72.88%	64.41%	50.28%	44.37%	57.29%	50.00%	53.40%
emotions	multi-label	16	-	70.06%	59.21%	50.28%	44.37%	51.79%	52.73%	52.25%
emotions	multi-label	16	1.02	71.19%	60.11%	53.67%	45.21%	53.57%	54.55%	54.05%
emotions	multi-label	16	1.04	70.34%	59.01%	50.28%	45.26%	52.34%	50.91%	51.61%
emotions	multi-label	16	1.08	73.73%	68.90%	51.13%	46.34%	58.76%	51.82%	55.07%
emotions	multi-label	16	1.16	72.03%	60.88%	48.02%	46.67%	55.56%	50.00%	52.63%
emotions	multi-label	16	1.32	75.14%	73.59%	51.41%	46.33%	62.79%	49.09%	55.10%
emotions	multi-label	32	-	74.86%	66.53%	55.65%	51.45%	60.19%	56.36%	58.22%
emotions	multi-label	32	1.02	73.45%	64.97%	56.50%	48.63%	57.41%	56.36%	56.88%
emotions	multi-label	32	1.04	73.16%	65.40%	52.26%	46.26%	57.28%	53.64%	55.40%
emotions	multi-label	32	1.08	73.16%	64.24%	51.41%	48.87%	57.58%	51.82%	54.55%
emotions	multi-label	32	1.16	70.34%	62.29%	49.15%	43.37%	52.43%	49.09%	50.70%
emotions	multi-label	32	1.32	72.60%	69.49%	48.31%	43.64%	56.99%	48.18%	52.22%
emotions	multi-label	64	-	74.01%	64.27%	57.91%	51.93%	58.18%	58.18%	58.18%
emotions	multi-label	64	1.02	74.58%	65.11%	57.91%	52.55%	59.26%	58.18%	58.72%
emotions	multi-label	64	1.04	73.45%	65.65%	55.65%	50.23%	57.55%	55.45%	56.48%
emotions	multi-label	64	1.08	72.88%	66.64%	51.13%	48.01%	56.86%	52.73%	54.72%
emotions	multi-label	64	1.16	72.88%	62.34%	53.11%	48.85%	56.73%	53.64%	55.14%
emotions	multi-label	64	1.32	70.06%	60.73%	47.18%	44.58%	51.96%	48.18%	50.00%
emotions	multi-label	128	-	73.73%	62.43%	52.54%	50.11%	58.59%	52.73%	55.50%
emotions	multi-label	128	1.02	73.16%	63.28%	51.69%	49.27%	57.58%	51.82%	54.55%
emotions	multi-label	128	1.04	74.01%	63.84%	51.13%	49.27%	59.57%	50.91%	54.90%
emotions	multi-label	128	1.08	74.86%	69.77%	55.08%	50.89%	61.05%	52.73%	56.59%
emotions	multi-label	128	1.16	75.42%	71.10%	55.37%	51.67%	62.11%	53.64%	57.56%
emotions	multi-label	128	1.32	72.88%	63.42%	57.34%	53.82%	56.48%	55.45%	55.96%
emotions	multi-label	256	-	74.58%	63.84%	58.76%	54.01%	59.43%	57.27%	58.33%
emotions	multi-label	256	1.02	74.01%	63.56%	55.93%	50.28%	58.65%	55.45%	57.01%
emotions	multi-label	256	1.04	73.73%	62.01%	58.47%	52.32%	57.66%	58.18%	57.92%

emotions	multi-label	256	1.08	75.71%	67.91%	64.12%	59.45%	60.34%	63.64%	61.95%
emotions	multi-label	256	1.16	73.16%	65.25%	56.78%	50.74%	57.28%	53.64%	55.40%
emotions	multi-label	256	1.32	76.27%	70.06%	57.06%	53.60%	62.75%	58.18%	60.38%
emotions	multi-label	512	-	71.19%	57.91%	48.31%	44.80%	53.85%	50.91%	52.34%
emotions	multi-label	512	1.02	73.16%	62.01%	57.63%	50.10%	57.01%	55.45%	56.22%
emotions	multi-label	512	1.04	75.42%	66.67%	48.87%	47.40%	63.22%	50.00%	55.84%
emotions	multi-label	512	1.08	72.03%	57.91%	50.28%	48.42%	55.34%	51.82%	53.52%
emotions	multi-label	512	1.16	74.01%	62.71%	59.04%	54.56%	58.04%	59.09%	58.56%
emotions	multi-label	512	1.32	72.88%	57.49%	58.76%	53.75%	55.74%	61.82%	58.62%

Table 9.: Performance of multi-label models learned from the data set yeast

parameters				performance measures						
data set	head ref.	m	l	Ha.Acc.	Ex.Prec.	Ex.Rec.	Ex.F1.	Mi.Prec.	Mi.Rec.	Mi.F1
yeast	multi-label	0	-	67.52%	52.81%	50.64%	46.85%	46.55%	51.03%	48.69%
yeast	multi-label	0	1.02	68.38%	54.59%	49.56%	46.36%	47.73%	49.46%	48.58%
yeast	multi-label	0	1.04	68.44%	53.99%	49.77%	46.47%	47.83%	49.75%	48.77%
yeast	multi-label	0	1.08	68.44%	54.17%	49.81%	46.49%	47.83%	49.75%	48.77%
yeast	multi-label	0	1.16	67.49%	53.12%	48.88%	45.60%	46.38%	49.07%	47.69%
yeast	multi-label	0	1.32	67.46%	53.24%	49.14%	45.48%	46.30%	48.58%	47.41%
yeast	multi-label	1	-	71.96%	56.37%	51.04%	49.58%	53.77%	51.03%	52.37%
yeast	multi-label	1	1.02	71.58%	56.34%	50.28%	48.95%	53.07%	50.83%	51.93%
yeast	multi-label	1	1.04	72.38%	57.76%	53.07%	50.98%	54.34%	53.48%	53.91%
yeast	multi-label	1	1.08	71.19%	55.60%	49.45%	48.09%	52.43%	49.75%	51.06%
yeast	multi-label	1	1.16	72.53%	56.24%	52.80%	50.74%	54.59%	53.68%	54.13%
yeast	multi-label	1	1.32	71.22%	53.56%	49.20%	47.49%	52.47%	50.05%	51.23%
yeast	multi-label	2	-	70.92%	55.07%	50.29%	48.86%	51.91%	50.64%	51.27%
yeast	multi-label	2	1.02	71.81%	56.42%	51.78%	50.28%	53.43%	52.01%	52.71%
yeast	multi-label	2	1.04	71.67%	55.75%	51.70%	49.68%	53.15%	52.11%	52.63%
yeast	multi-label	2	1.08	71.90%	55.68%	50.25%	48.90%	53.67%	50.93%	52.27%
yeast	multi-label	2	1.16	71.96%	55.20%	51.48%	50.40%	53.62%	52.99%	53.31%
yeast	multi-label	2	1.32	71.70%	54.94%	53.78%	50.83%	53.09%	53.88%	53.48%
yeast	multi-label	4	-	72.17%	56.65%	52.30%	51.10%	54.03%	52.60%	53.31%
yeast	multi-label	4	1.02	73.03%	58.34%	53.33%	51.51%	55.54%	53.58%	54.55%
yeast	multi-label	4	1.04	72.88%	56.62%	51.56%	50.80%	55.37%	52.60%	53.95%
yeast	multi-label	4	1.08	72.91%	57.13%	53.04%	51.11%	55.31%	53.68%	54.48%
yeast	multi-label	4	1.16	72.38%	55.94%	52.93%	51.14%	54.37%	53.09%	53.72%
yeast	multi-label	4	1.32	71.84%	56.42%	53.34%	50.31%	53.42%	52.89%	53.16%
yeast	multi-label	8	-	72.79%	57.88%	52.82%	51.32%	55.13%	53.29%	54.19%
yeast	multi-label	8	1.02	73.24%	59.11%	52.42%	51.01%	55.98%	53.29%	54.60%
yeast	multi-label	8	1.04	73.09%	58.50%	52.35%	50.74%	55.70%	53.19%	54.42%
yeast	multi-label	8	1.08	73.59%	59.78%	54.23%	52.37%	56.54%	54.27%	55.38%
yeast	multi-label	8	1.16	72.29%	55.66%	53.59%	50.76%	54.14%	53.88%	54.01%
yeast	multi-label	8	1.32	71.40%	57.04%	53.82%	51.19%	52.61%	53.39%	53.00%
yeast	multi-label	16	-	72.91%	58.83%	53.94%	51.92%	55.37%	53.09%	54.21%
yeast	multi-label	16	1.02	73.06%	59.51%	54.10%	51.88%	55.57%	53.88%	54.71%
yeast	multi-label	16	1.04	73.33%	57.75%	52.28%	50.58%	56.32%	52.01%	54.08%

yeast	multi-label	16	1.08	72.58%	57.37%	53.69%	51.01%	54.61%	54.66%	54.63%
yeast	multi-label	16	1.16	72.14%	55.71%	52.73%	50.00%	53.97%	52.70%	53.33%
yeast	multi-label	16	1.32	71.93%	57.51%	53.50%	50.36%	53.53%	53.58%	53.56%
yeast	multi-label	32	-	71.81%	57.25%	50.93%	50.08%	53.43%	52.01%	52.71%
yeast	multi-label	32	1.02	72.79%	57.80%	53.78%	51.75%	55.03%	54.17%	54.60%
yeast	multi-label	32	1.04	72.97%	58.44%	53.92%	52.28%	55.43%	53.58%	54.49%
yeast	multi-label	32	1.08	72.44%	56.49%	52.80%	50.85%	54.43%	53.68%	54.05%
yeast	multi-label	32	1.16	72.26%	54.85%	54.86%	51.47%	54.03%	54.66%	54.34%
yeast	multi-label	32	1.32	72.23%	56.38%	53.29%	50.27%	54.04%	53.78%	53.91%
yeast	multi-label	64	-	73.80%	59.86%	55.45%	53.95%	56.84%	55.05%	55.93%
yeast	multi-label	64	1.02	74.99%	62.06%	55.42%	54.70%	59.16%	55.45%	57.24%
yeast	multi-label	64	1.04	72.97%	58.19%	54.17%	52.57%	55.37%	54.17%	54.76%
yeast	multi-label	64	1.08	73.00%	58.60%	55.50%	53.11%	55.26%	55.64%	55.45%
yeast	multi-label	64	1.16	72.70%	56.34%	55.88%	52.70%	54.67%	56.33%	55.49%
yeast	multi-label	64	1.32	72.67%	56.79%	54.36%	51.24%	54.79%	54.47%	54.63%
yeast	multi-label	128	-	73.30%	57.67%	54.56%	52.53%	55.98%	54.17%	55.06%
yeast	multi-label	128	1.02	72.70%	56.98%	57.05%	53.66%	54.56%	57.51%	56.00%
yeast	multi-label	128	1.04	74.01%	59.54%	58.01%	54.85%	56.91%	57.41%	57.16%
yeast	multi-label	128	1.08	73.59%	57.98%	54.26%	52.72%	56.41%	55.25%	55.83%
yeast	multi-label	128	1.16	74.33%	57.77%	56.98%	54.28%	57.46%	57.80%	57.63%
yeast	multi-label	128	1.32	71.84%	56.38%	53.24%	50.88%	53.41%	53.09%	53.25%
yeast	multi-label	256	-	72.70%	56.29%	55.24%	51.69%	54.79%	54.96%	54.88%
yeast	multi-label	256	1.02	71.96%	54.34%	55.54%	51.65%	53.40%	56.33%	54.82%
yeast	multi-label	256	1.04	73.06%	57.10%	54.66%	51.70%	55.50%	54.47%	54.98%
yeast	multi-label	256	1.08	73.15%	57.43%	59.19%	55.35%	55.13%	59.57%	57.26%
yeast	multi-label	256	1.16	73.47%	57.60%	55.04%	53.01%	56.28%	54.56%	55.41%
yeast	multi-label	256	1.32	72.44%	57.65%	56.09%	52.33%	54.32%	54.86%	54.59%
yeast	multi-label	512	-	72.94%	57.79%	57.81%	54.26%	54.97%	57.51%	56.21%
yeast	multi-label	512	1.02	72.61%	55.92%	57.52%	53.37%	54.44%	57.21%	55.79%
yeast	multi-label	512	1.04	72.94%	56.40%	54.64%	51.93%	55.28%	54.47%	54.87%
yeast	multi-label	512	1.08	72.29%	57.20%	54.24%	51.68%	54.10%	54.37%	54.23%
yeast	multi-label	512	1.16	72.88%	56.43%	54.97%	52.38%	55.16%	54.56%	54.86%
yeast	multi-label	512	1.32	72.70%	55.74%	55.60%	52.00%	54.68%	56.23%	55.44%

Table 10.: Performance of multi-label models learned from the data set medical

parameters				performance measures						
data set	head ref.	m	l	Ha.Acc.	Ex.Prec.	Ex.Rec.	Ex.F1.	Mi.Prec.	Mi.Rec.	Mi.F1
medical	multi-label	0	-	97.46%	69.44%	78.97%	64.30%	53.01%	79.18%	63.50%
medical	multi-label	0	1.02	97.23%	67.93%	76.67%	61.28%	50.27%	77.14%	60.87%
medical	multi-label	0	1.04	97.23%	67.93%	76.67%	61.28%	50.27%	77.14%	60.87%
medical	multi-label	0	1.08	97.23%	67.93%	76.67%	61.28%	50.27%	77.14%	60.87%
medical	multi-label	0	1.16	97.24%	68.10%	76.67%	61.38%	50.40%	77.14%	60.97%
medical	multi-label	0	1.32	97.26%	68.11%	77.44%	61.96%	50.66%	77.96%	61.41%
medical	multi-label	1	-	98.10%	76.10%	75.90%	66.90%	63.27%	75.92%	69.02%
medical	multi-label	1	1.02	98.12%	76.10%	76.41%	67.41%	63.51%	76.73%	69.50%
medical	multi-label	1	1.04	98.10%	75.85%	76.41%	67.24%	63.09%	76.73%	69.24%

medical	multi-label	1	1.08	98.14%	76.27%	77.18%	68.17%	63.76%	77.55%	69.98%
medical	multi-label	1	1.16	98.11%	75.98%	76.15%	67.24%	63.30%	76.73%	69.37%
medical	multi-label	1	1.32	98.11%	76.07%	75.90%	67.04%	63.30%	76.73%	69.37%
medical	multi-label	2	-	98.29%	78.12%	75.64%	68.31%	67.27%	75.51%	71.15%
medical	multi-label	2	1.02	98.28%	77.86%	75.13%	67.97%	67.15%	75.10%	70.91%
medical	multi-label	2	1.04	98.28%	77.86%	75.13%	67.97%	67.15%	75.10%	70.91%
medical	multi-label	2	1.08	98.36%	78.89%	76.67%	69.40%	68.23%	77.14%	72.41%
medical	multi-label	2	1.16	98.31%	78.97%	75.64%	68.63%	67.64%	75.92%	71.54%
medical	multi-label	2	1.32	98.37%	80.04%	76.67%	69.49%	68.48%	77.14%	72.55%
medical	multi-label	4	-	98.53%	81.79%	76.41%	71.18%	72.14%	77.14%	74.56%
medical	multi-label	4	1.02	98.54%	81.79%	76.92%	71.69%	72.24%	77.55%	74.80%
medical	multi-label	4	1.04	98.53%	81.79%	76.41%	71.18%	72.14%	77.14%	74.56%
medical	multi-label	4	1.08	98.55%	82.82%	75.90%	71.01%	72.87%	76.73%	74.75%
medical	multi-label	4	1.16	98.45%	82.22%	74.36%	68.96%	71.04%	75.10%	73.02%
medical	multi-label	4	1.32	98.30%	80.06%	75.38%	69.49%	67.39%	75.92%	71.40%
medical	multi-label	8	-	98.77%	84.79%	79.49%	74.65%	76.65%	80.41%	78.49%
medical	multi-label	8	1.02	98.78%	84.53%	79.23%	74.75%	77.17%	80.00%	78.56%
medical	multi-label	8	1.04	98.79%	84.53%	79.49%	74.92%	77.25%	80.41%	78.80%
medical	multi-label	8	1.08	98.78%	84.36%	79.23%	74.58%	77.17%	80.00%	78.56%
medical	multi-label	8	1.16	98.81%	85.31%	80.77%	75.92%	77.43%	81.22%	79.28%
medical	multi-label	8	1.32	98.45%	83.02%	77.95%	73.06%	69.53%	79.18%	74.05%
medical	multi-label	16	-	98.85%	85.64%	80.51%	77.33%	78.80%	80.41%	79.60%
medical	multi-label	16	1.02	98.84%	85.56%	79.23%	76.50%	79.18%	79.18%	79.18%
medical	multi-label	16	1.04	98.84%	85.04%	79.23%	76.50%	79.18%	79.18%	79.18%
medical	multi-label	16	1.08	98.75%	84.54%	80.00%	76.05%	76.26%	80.00%	78.09%
medical	multi-label	16	1.16	98.62%	85.25%	78.97%	74.24%	73.66%	78.78%	76.13%
medical	multi-label	16	1.32	98.44%	84.68%	75.90%	71.89%	70.45%	75.92%	73.08%
medical	multi-label	32	-	98.89%	87.44%	79.49%	76.91%	81.09%	78.78%	79.92%
medical	multi-label	32	1.02	98.89%	87.09%	79.23%	76.75%	81.09%	78.78%	79.92%
medical	multi-label	32	1.04	98.84%	86.51%	79.23%	76.74%	79.42%	78.78%	79.10%
medical	multi-label	32	1.08	98.72%	84.78%	79.23%	76.02%	76.49%	78.37%	77.42%
medical	multi-label	32	1.16	98.67%	85.32%	78.97%	75.29%	75.20%	77.96%	76.55%
medical	multi-label	32	1.32	98.14%	82.23%	80.26%	73.83%	63.40%	79.18%	70.42%
medical	multi-label	64	-	98.97%	87.69%	81.03%	79.13%	82.98%	79.59%	81.25%
medical	multi-label	64	1.02	98.88%	86.36%	80.51%	78.28%	80.50%	79.18%	79.84%
medical	multi-label	64	1.04	98.92%	87.47%	80.00%	78.09%	81.78%	78.78%	80.25%
medical	multi-label	64	1.08	98.89%	86.80%	80.51%	77.54%	80.58%	79.59%	80.08%
medical	multi-label	64	1.16	98.70%	85.49%	78.46%	75.63%	76.73%	76.73%	76.73%
medical	multi-label	64	1.32	98.67%	86.19%	80.51%	77.39%	75.00%	78.37%	76.65%
medical	multi-label	128	-	99.04%	87.86%	84.36%	81.69%	82.59%	83.27%	82.93%
medical	multi-label	128	1.02	98.92%	86.41%	82.31%	79.33%	80.49%	80.82%	80.65%
medical	multi-label	128	1.04	99.01%	87.78%	82.56%	80.55%	83.19%	80.82%	81.99%
medical	multi-label	128	1.08	98.80%	85.40%	81.28%	78.50%	78.00%	79.59%	78.79%
medical	multi-label	128	1.16	98.34%	82.89%	83.85%	77.49%	66.34%	82.04%	73.36%
medical	multi-label	128	1.32	98.52%	84.39%	80.51%	75.93%	71.06%	79.18%	74.90%
medical	multi-label	256	-	99.00%	85.90%	84.10%	81.33%	81.53%	82.86%	82.19%
medical	multi-label	256	1.02	98.93%	85.98%	82.31%	79.86%	81.33%	80.00%	80.66%

medical	multi-label	256	1.04	98.77%	84.36%	79.49%	76.99%	78.42%	77.14%	77.78%
medical	multi-label	256	1.08	98.85%	84.69%	81.28%	78.37%	78.80%	80.41%	79.60%
medical	multi-label	256	1.16	98.25%	80.89%	82.82%	74.21%	64.72%	81.63%	72.20%
medical	multi-label	256	1.32	97.54%	76.92%	82.82%	72.21%	53.95%	80.82%	64.71%
medical	multi-label	512	-	99.02%	85.30%	85.13%	82.12%	81.42%	84.08%	82.73%
medical	multi-label	512	1.02	98.78%	85.30%	81.79%	78.09%	77.17%	80.00%	78.56%
medical	multi-label	512	1.04	98.87%	86.49%	80.00%	77.81%	80.67%	78.37%	79.50%
medical	multi-label	512	1.08	98.83%	85.51%	78.21%	76.63%	80.60%	76.33%	78.41%
medical	multi-label	512	1.16	98.04%	79.32%	80.26%	73.38%	61.66%	78.78%	69.18%
medical	multi-label	512	1.32	97.97%	79.04%	79.49%	71.79%	60.57%	78.37%	68.33%

Table 11.: Performance of multi-label models learned from the data set scene

parameters				performance measures						
data set	head ref.	m	l	Ha.Acc.	Ex.Prec.	Ex.Rec.	Ex.F1.	Mi.Prec.	Mi.Rec.	Mi.F1
scene	multi-label	0	-	77.43%	60.20%	56.25%	41.02%	40.50%	56.42%	47.15%
scene	multi-label	0	1.02	78.12%	65.76%	53.54%	39.96%	41.32%	53.70%	46.70%
scene	multi-label	0	1.04	78.47%	66.04%	53.96%	40.21%	42.04%	54.47%	47.46%
scene	multi-label	0	1.08	79.03%	66.00%	56.87%	43.09%	43.36%	57.20%	49.33%
scene	multi-label	0	1.16	77.43%	63.41%	55.42%	40.96%	40.40%	55.64%	46.81%
scene	multi-label	0	1.32	77.08%	61.87%	57.92%	42.01%	40.11%	57.59%	47.28%
scene	multi-label	1	-	80.49%	66.62%	51.46%	40.10%	45.86%	51.75%	48.63%
scene	multi-label	1	1.02	80.28%	66.19%	53.33%	41.38%	45.54%	53.70%	49.29%
scene	multi-label	1	1.04	80.35%	66.75%	52.92%	41.18%	45.67%	53.31%	49.19%
scene	multi-label	1	1.08	80.21%	64.44%	56.67%	44.70%	45.62%	56.81%	50.61%
scene	multi-label	1	1.16	81.60%	67.58%	58.33%	45.61%	48.70%	58.37%	53.10%
scene	multi-label	1	1.32	80.83%	66.42%	54.79%	42.99%	46.86%	55.25%	50.71%
scene	multi-label	2	-	80.83%	66.18%	54.58%	43.29%	46.86%	55.25%	50.71%
scene	multi-label	2	1.02	80.90%	66.18%	55.83%	43.73%	47.08%	56.42%	51.33%
scene	multi-label	2	1.04	80.62%	65.94%	54.58%	43.19%	46.41%	55.25%	50.44%
scene	multi-label	2	1.08	80.90%	66.81%	55.83%	44.38%	47.08%	56.42%	51.33%
scene	multi-label	2	1.16	81.81%	67.43%	58.13%	47.26%	49.18%	58.37%	53.38%
scene	multi-label	2	1.32	81.32%	68.99%	55.21%	44.03%	47.97%	55.25%	51.36%
scene	multi-label	4	-	82.15%	69.55%	55.00%	43.75%	50.00%	55.25%	52.50%
scene	multi-label	4	1.02	81.94%	68.54%	54.58%	43.51%	49.47%	54.86%	52.03%
scene	multi-label	4	1.04	82.01%	68.51%	54.58%	43.56%	49.65%	54.86%	52.13%
scene	multi-label	4	1.08	81.60%	67.03%	54.58%	43.68%	48.64%	55.64%	51.91%
scene	multi-label	4	1.16	81.04%	65.35%	52.71%	42.74%	47.24%	53.31%	50.09%
scene	multi-label	4	1.32	82.64%	68.69%	58.13%	47.78%	51.21%	57.59%	54.21%
scene	multi-label	8	-	81.88%	66.70%	52.29%	43.65%	49.28%	53.31%	51.21%
scene	multi-label	8	1.02	82.36%	68.07%	55.83%	46.83%	50.52%	56.81%	53.48%
scene	multi-label	8	1.04	82.85%	68.28%	57.08%	48.51%	51.74%	57.98%	54.68%
scene	multi-label	8	1.08	82.43%	67.12%	55.00%	45.96%	50.71%	55.25%	52.89%
scene	multi-label	8	1.16	82.43%	66.35%	57.29%	47.32%	50.68%	57.59%	53.92%
scene	multi-label	8	1.32	82.71%	69.73%	58.96%	48.82%	51.36%	58.75%	54.81%
scene	multi-label	16	-	84.72%	72.26%	58.96%	51.06%	56.98%	58.75%	57.85%
scene	multi-label	16	1.02	84.72%	72.34%	58.54%	50.72%	57.03%	58.37%	57.69%

scene	multi-label	16	1.04	84.51%	71.18%	59.79%	51.37%	56.30%	59.14%	57.69%
scene	multi-label	16	1.08	84.79%	71.15%	59.58%	51.25%	57.14%	59.14%	58.13%
scene	multi-label	16	1.16	83.47%	67.08%	61.67%	51.60%	53.22%	61.09%	56.88%
scene	multi-label	16	1.32	82.43%	68.12%	55.21%	46.62%	50.71%	55.25%	52.89%
scene	multi-label	32	-	83.47%	67.01%	57.71%	49.44%	53.41%	57.98%	55.60%
scene	multi-label	32	1.02	83.82%	66.35%	59.38%	51.29%	54.29%	59.14%	56.61%
scene	multi-label	32	1.04	84.58%	69.86%	59.17%	51.15%	56.51%	59.14%	57.79%
scene	multi-label	32	1.08	84.93%	70.38%	62.08%	52.99%	57.19%	61.87%	59.44%
scene	multi-label	32	1.16	83.89%	68.76%	57.29%	48.93%	54.61%	57.59%	56.06%
scene	multi-label	32	1.32	85.21%	71.01%	60.21%	53.11%	58.33%	59.92%	59.12%
scene	multi-label	64	-	85.97%	74.24%	63.96%	56.01%	60.22%	63.04%	61.60%
scene	multi-label	64	1.02	85.97%	73.99%	63.75%	56.00%	60.30%	62.65%	61.45%
scene	multi-label	64	1.04	86.18%	73.40%	64.17%	56.53%	60.74%	63.81%	62.24%
scene	multi-label	64	1.08	84.86%	70.35%	62.50%	54.31%	56.94%	62.26%	59.48%
scene	multi-label	64	1.16	86.04%	72.47%	65.42%	57.17%	60.07%	64.98%	62.43%
scene	multi-label	64	1.32	84.17%	69.70%	59.58%	51.38%	55.23%	59.53%	57.30%
scene	multi-label	128	-	86.11%	72.64%	63.54%	56.04%	60.52%	63.81%	62.12%
scene	multi-label	128	1.02	86.18%	72.29%	65.21%	57.01%	60.43%	65.37%	62.80%
scene	multi-label	128	1.04	85.83%	72.78%	64.79%	56.57%	59.50%	64.59%	61.94%
scene	multi-label	128	1.08	86.53%	73.37%	64.58%	57.47%	61.54%	65.37%	63.40%
scene	multi-label	128	1.16	84.65%	72.19%	59.79%	51.83%	56.72%	59.14%	57.90%
scene	multi-label	128	1.32	85.83%	74.27%	58.54%	53.14%	60.91%	57.59%	59.20%
scene	multi-label	256	-	85.56%	69.96%	64.38%	57.04%	58.72%	64.20%	61.34%
scene	multi-label	256	1.02	85.83%	69.72%	66.25%	58.64%	59.36%	65.37%	62.22%
scene	multi-label	256	1.04	86.74%	73.65%	65.42%	58.32%	62.41%	64.59%	63.48%
scene	multi-label	256	1.08	87.08%	75.99%	63.12%	57.88%	64.14%	62.65%	63.39%
scene	multi-label	256	1.16	86.81%	75.97%	60.83%	56.54%	63.79%	60.31%	62.00%
scene	multi-label	256	1.32	87.01%	77.92%	57.92%	53.75%	65.62%	57.20%	61.12%
scene	multi-label	512	-	86.18%	70.97%	67.50%	59.97%	60.28%	66.15%	63.08%
scene	multi-label	512	1.02	86.88%	73.40%	64.79%	58.67%	63.08%	63.81%	63.44%
scene	multi-label	512	1.04	86.04%	72.67%	65.00%	57.39%	60.29%	63.81%	62.00%
scene	multi-label	512	1.08	87.64%	75.38%	69.17%	62.60%	64.58%	68.09%	66.29%
scene	multi-label	512	1.16	87.71%	75.56%	66.04%	59.97%	65.62%	65.37%	65.50%
scene	multi-label	512	1.32	85.97%	77.85%	60.42%	53.39%	60.96%	59.53%	60.24%

Table 12.: Performance of single-label models learned from the data set birds

parameters			performance measures						
data set	head ref.	m	Ha.Acc.	Ex.Prec.	Ex.Rec.	Ex.F1.	Mi.Prec.	Mi.Rec.	Mi.F1
birds	single-label	0	92.35%	69.74%	61.88%	55.12%	32.43%	35.82%	34.04%
birds	single-label	1	93.50%	73.75%	64.22%	57.77%	41.18%	41.79%	41.48%
birds	single-label	2	93.91%	72.58%	67.08%	57.59%	44.93%	46.27%	45.59%
birds	single-label	4	94.16%	71.93%	66.56%	57.30%	46.97%	46.27%	46.62%
birds	single-label	8	94.33%	74.90%	65.91%	57.70%	48.48%	47.76%	48.12%
birds	single-label	16	94.90%	78.67%	65.60%	61.59%	54.10%	49.25%	51.56%
birds	single-label	32	94.90%	75.55%	65.21%	57.48%	54.10%	49.25%	51.56%
birds	single-label	64	94.65%	72.14%	67.81%	58.14%	51.61%	47.76%	49.61%

birds	single-label	128	94.08%	63.65%	67.55%	54.72%	46.48%	49.25%	47.83%
birds	single-label	256	94.33%	71.64%	67.55%	58.29%	48.48%	47.76%	48.12%
birds	single-label	512	94.24%	71.15%	68.78%	58.66%	47.95%	52.24%	50.00%

Table 13.: Performance of single-label models learned from the data set emotions

parameters			performance measures						
data set	head ref.	m	Ha.Acc.	Ex.Prec.	Ex.Rec.	Ex.F1.	Mi.Prec.	Mi.Rec.	Mi.F1
emotions	single-label	0	66.95%	64.52%	50.28%	39.98%	46.90%	48.18%	47.53%
emotions	single-label	1	68.93%	66.07%	48.87%	42.77%	50.00%	48.18%	49.07%
emotions	single-label	2	70.06%	69.75%	54.24%	44.87%	51.75%	53.64%	52.68%
emotions	single-label	4	71.19%	70.14%	55.08%	47.03%	53.77%	51.82%	52.78%
emotions	single-label	8	73.45%	71.38%	53.11%	50.54%	58.00%	52.73%	55.24%
emotions	single-label	16	70.06%	59.21%	50.28%	44.37%	51.79%	52.73%	52.25%
emotions	single-label	32	74.86%	66.53%	55.65%	51.45%	60.19%	56.36%	58.22%
emotions	single-label	64	74.01%	64.27%	57.91%	51.93%	58.18%	58.18%	58.18%
emotions	single-label	128	73.73%	62.43%	52.54%	50.11%	58.59%	52.73%	55.50%
emotions	single-label	256	74.58%	63.84%	58.76%	54.01%	59.43%	57.27%	58.33%
emotions	single-label	512	71.19%	57.91%	48.31%	44.80%	53.85%	50.91%	52.34%

Table 14.: Performance of single-label models learned from the data set yeast

parameters			performance measures						
data set	head ref.	m	Ha.Acc.	Ex.Prec.	Ex.Rec.	Ex.F1.	Mi.Prec.	Mi.Rec.	Mi.F1
yeast	single-label	0	67.52%	52.81%	50.64%	46.85%	46.55%	51.03%	48.69%
yeast	single-label	1	71.93%	56.23%	51.04%	49.56%	53.72%	51.03%	52.34%
yeast	single-label	2	70.90%	55.07%	50.29%	48.87%	51.86%	50.64%	51.24%
yeast	single-label	4	72.17%	56.65%	52.30%	51.10%	54.03%	52.60%	53.31%
yeast	single-label	8	72.79%	57.88%	52.82%	51.32%	55.13%	53.29%	54.19%
yeast	single-label	16	72.91%	58.83%	53.94%	51.92%	55.37%	53.09%	54.21%
yeast	single-label	32	71.81%	57.25%	50.93%	50.08%	53.43%	52.01%	52.71%
yeast	single-label	64	73.80%	59.86%	55.45%	53.95%	56.84%	55.05%	55.93%
yeast	single-label	128	73.30%	57.67%	54.56%	52.53%	55.98%	54.17%	55.06%
yeast	single-label	256	72.70%	56.29%	55.24%	51.69%	54.79%	54.96%	54.88%
yeast	single-label	512	72.94%	57.79%	57.81%	54.26%	54.97%	57.51%	56.21%

Table 15.: Performance of single-label models learned from the data set medical

parameters			performance measures						
data set	head ref.	m	Ha.Acc.	Ex.Prec.	Ex.Rec.	Ex.F1.	Mi.Prec.	Mi.Rec.	Mi.F1
medical	single-label	0	97.46%	69.44%	78.97%	64.30%	53.01%	79.18%	63.50%
medical	single-label	1	98.10%	76.10%	75.90%	66.90%	63.27%	75.92%	69.02%
medical	single-label	2	98.29%	78.12%	75.64%	68.31%	67.27%	75.51%	71.15%
medical	single-label	4	98.53%	81.79%	76.41%	71.18%	72.14%	77.14%	74.56%
medical	single-label	8	98.77%	84.79%	79.49%	74.65%	76.65%	80.41%	78.49%
medical	single-label	16	98.85%	85.64%	80.51%	77.33%	78.80%	80.41%	79.60%

medical	single-label	32	98.89%	87.44%	79.49%	76.91%	81.09%	78.78%	79.92%
medical	single-label	64	98.97%	87.69%	81.03%	79.13%	82.98%	79.59%	81.25%
medical	single-label	128	99.04%	87.86%	84.36%	81.69%	82.59%	83.27%	82.93%
medical	single-label	256	99.00%	85.90%	84.10%	81.33%	81.53%	82.86%	82.19%
medical	single-label	512	99.02%	85.30%	85.13%	82.12%	81.42%	84.08%	82.73%

Table 16.: Performance of single-label models learned from the data set scene

parameters			performance measures						
data set	head ref.	m	Ha.Acc.	Ex.Prec.	Ex.Rec.	Ex.F1.	Mi.Prec.	Mi.Rec.	Mi.F1
scene	single-label	0	77.43%	60.20%	56.25%	41.02%	40.50%	56.42%	47.15%
scene	single-label	1	80.49%	66.62%	51.46%	40.10%	45.86%	51.75%	48.63%
scene	single-label	2	80.83%	66.18%	54.58%	43.29%	46.86%	55.25%	50.71%
scene	single-label	4	82.15%	69.55%	55.00%	43.75%	50.00%	55.25%	52.50%
scene	single-label	8	81.88%	66.70%	52.29%	43.65%	49.28%	53.31%	51.21%
scene	single-label	16	84.72%	72.26%	58.96%	51.06%	56.98%	58.75%	57.85%
scene	single-label	32	83.47%	67.01%	57.71%	49.44%	53.41%	57.98%	55.60%
scene	single-label	64	85.97%	74.24%	63.96%	56.01%	60.22%	63.04%	61.60%
scene	single-label	128	86.11%	72.64%	63.54%	56.04%	60.52%	63.81%	62.12%
scene	single-label	256	85.56%	69.96%	64.38%	57.04%	58.72%	64.20%	61.34%
scene	single-label	512	86.18%	70.97%	67.50%	59.97%	60.28%	66.15%	63.08%

## B. Model Characteristics

Table 17.: Model characteristics of multi-label models learned from the data set birds

data set	parameters			model characteristics					runtime
	head ref.	m	l	#rules	%MLHR	Avg.Cond.	Avg.MLH	opt. #rules	runtime
birds	multi-label	0	-	518	0.00%	1.53	-	286	0:27:25
birds	multi-label	0	1.02	370	49.46%	1.58	2.00	16	0:19:18
birds	multi-label	0	1.04	370	49.46%	1.58	2.00	16	0:19:32
birds	multi-label	0	1.08	370	49.46%	1.58	2.00	16	0:19:24
birds	multi-label	0	1.16	370	49.46%	1.58	2.00	16	0:19:34
birds	multi-label	0	1.32	369	49.32%	1.58	2.00	16	0:19:40
birds	multi-label	1	-	446	0.00%	2.06	-	6	0:23:28
birds	multi-label	1	1.02	397	11.08%	2.08	2.11	6	0:20:22
birds	multi-label	1	1.04	392	11.99%	2.09	2.15	6	0:20:06
birds	multi-label	1	1.08	386	13.47%	2.09	2.13	6	0:19:41
birds	multi-label	1	1.16	372	18.28%	2.09	2.15	31	0:18:50
birds	multi-label	1	1.32	366	27.05%	2.18	2.09	26	0:19:14
birds	multi-label	2	-	417	0.48%	2.17	3.00	51	0:21:59
birds	multi-label	2	1.02	386	8.81%	2.18	2.15	6	0:20:01
birds	multi-label	2	1.04	386	8.81%	2.18	2.15	6	0:20:01
birds	multi-label	2	1.08	385	8.05%	2.19	2.23	6	0:20:25
birds	multi-label	2	1.16	381	9.71%	2.20	2.22	46	0:20:09
birds	multi-label	2	1.32	357	19.05%	2.25	2.28	16	0:18:17
birds	multi-label	4	-	393	0.00%	2.35	-	6	0:20:24
birds	multi-label	4	1.02	365	6.58%	2.35	2.04	6	0:19:40
birds	multi-label	4	1.04	366	6.56%	2.35	2.04	6	0:19:35
birds	multi-label	4	1.08	367	6.54%	2.35	2.04	6	0:18:47
birds	multi-label	4	1.16	362	10.22%	2.44	2.16	6	0:18:16
birds	multi-label	4	1.32	352	21.02%	2.61	2.55	11	0:17:58
birds	multi-label	8	-	355	0.00%	2.54	-	6	0:19:40
birds	multi-label	8	1.02	337	5.64%	2.55	2.11	6	0:17:46
birds	multi-label	8	1.04	334	6.59%	2.57	2.09	6	0:17:49
birds	multi-label	8	1.08	332	7.53%	2.61	2.24	6	0:18:04
birds	multi-label	8	1.16	322	13.35%	2.75	2.49	6	0:17:55
birds	multi-label	8	1.32	327	25.08%	2.98	2.87	6	0:17:30
birds	multi-label	16	-	322	0.31%	2.78	3.00	6	0:17:10
birds	multi-label	16	1.02	313	4.47%	2.81	2.07	6	0:16:26
birds	multi-label	16	1.04	316	5.70%	2.87	2.11	6	0:17:03
birds	multi-label	16	1.08	323	6.81%	2.98	2.32	6	0:17:11

birds	multi-label	16	1.16	325	18.15%	3.11	2.90	6	0:17:28
birds	multi-label	16	1.32	322	24.84%	3.27	3.11	6	0:17:49
birds	multi-label	32	-	308	0.00%	3.01	-	6	0:16:43
birds	multi-label	32	1.02	300	3.33%	3.06	2.00	6	0:15:50
birds	multi-label	32	1.04	299	5.69%	3.18	2.18	6	0:15:46
birds	multi-label	32	1.08	302	13.91%	3.32	2.90	6	0:16:52
birds	multi-label	32	1.16	305	15.41%	3.51	3.00	6	0:16:50
birds	multi-label	32	1.32	312	20.19%	3.75	3.02	6	0:17:14
birds	multi-label	64	-	288	0.00%	3.39	-	11	0:15:55
birds	multi-label	64	1.02	283	3.53%	3.45	2.20	11	0:15:24
birds	multi-label	64	1.04	276	12.32%	3.62	2.85	11	0:15:39
birds	multi-label	64	1.08	270	11.85%	3.81	3.03	11	0:16:38
birds	multi-label	64	1.16	278	13.67%	4.18	3.16	11	0:17:14
birds	multi-label	64	1.32	319	20.06%	4.50	3.03	11	0:18:21
birds	multi-label	128	-	249	0.00%	3.89	-	11	0:15:32
birds	multi-label	128	1.02	268	5.22%	4.15	2.86	11	0:15:33
birds	multi-label	128	1.04	257	9.34%	4.27	2.92	11	0:16:22
birds	multi-label	128	1.08	269	10.04%	4.63	2.96	16	0:17:46
birds	multi-label	128	1.16	285	9.82%	5.00	3.29	6	0:19:16
birds	multi-label	128	1.32	292	24.32%	5.41	2.94	11	0:20:59
birds	multi-label	256	-	247	0.00%	4.47	-	6	0:15:53
birds	multi-label	256	1.02	240	5.83%	4.80	2.71	6	0:17:04
birds	multi-label	256	1.04	263	6.46%	5.16	3.06	11	0:19:09
birds	multi-label	256	1.08	280	8.57%	5.35	3.21	11	0:19:52
birds	multi-label	256	1.16	282	10.99%	5.56	3.16	6	0:21:05
birds	multi-label	256	1.32	317	20.82%	5.95	2.95	16	0:24:29
birds	multi-label	512	-	237	0.00%	4.67	-	6	0:16:29
birds	multi-label	512	1.02	273	4.76%	5.29	3.23	6	0:18:26
birds	multi-label	512	1.04	251	5.98%	5.43	3.07	6	0:20:22
birds	multi-label	512	1.08	258	10.08%	5.75	3.23	6	0:21:54
birds	multi-label	512	1.16	305	7.87%	6.10	3.21	16	0:24:12
birds	multi-label	512	1.32	333	20.72%	6.37	2.94	11	0:26:48

Table 18.: Model characteristics of multi-label models learned from the data set emotions

parameters			model characteristics					runtime	
data set	head ref.	m	l	#rules	%MLHR	Avg.Cond.	Avg.MLH	opt. #rules	runtime
emotions	multi-label	0	-	841	0.00%	1.74	-	241	0:11:47
emotions	multi-label	0	1.02	550	64.18%	1.81	2.00	181	0:07:36
emotions	multi-label	0	1.04	550	64.18%	1.81	2.00	181	0:07:59
emotions	multi-label	0	1.08	550	64.18%	1.81	2.00	181	0:07:58
emotions	multi-label	0	1.16	546	64.29%	1.80	2.00	181	0:07:27
emotions	multi-label	0	1.32	541	64.70%	1.80	2.00	181	0:07:49
emotions	multi-label	1	-	509	0.00%	2.58	-	231	0:07:03
emotions	multi-label	1	1.02	467	10.28%	2.59	2.00	166	0:06:27
emotions	multi-label	1	1.04	471	12.74%	2.62	2.00	176	0:06:37
emotions	multi-label	1	1.08	459	14.38%	2.62	2.00	6	0:06:21

emotions	multi-label	1	1.16	449	23.16%	2.62	2.00	6	0:06:16
emotions	multi-label	1	1.32	436	36.47%	2.67	2.00	6	0:06:39
emotions	multi-label	2	-	439	0.00%	2.73	-	6	0:06:16
emotions	multi-label	2	1.02	429	8.62%	2.72	2.00	6	0:05:59
emotions	multi-label	2	1.04	440	8.86%	2.71	2.00	6	0:06:09
emotions	multi-label	2	1.08	430	11.16%	2.73	2.00	6	0:06:09
emotions	multi-label	2	1.16	413	15.98%	2.78	2.02	6	0:06:06
emotions	multi-label	2	1.32	421	21.85%	2.82	2.01	6	0:05:51
emotions	multi-label	4	-	399	0.00%	2.88	-	6	0:05:30
emotions	multi-label	4	1.02	389	6.43%	2.89	2.00	66	0:05:32
emotions	multi-label	4	1.04	386	6.48%	2.89	2.00	106	0:05:19
emotions	multi-label	4	1.08	366	9.56%	2.93	2.00	6	0:05:27
emotions	multi-label	4	1.16	362	10.50%	2.95	2.03	6	0:05:26
emotions	multi-label	4	1.32	359	18.38%	3.12	2.03	6	0:04:59
emotions	multi-label	8	-	334	0.00%	3.12	-	116	0:04:54
emotions	multi-label	8	1.02	316	5.06%	3.10	2.00	151	0:04:35
emotions	multi-label	8	1.04	314	3.50%	3.11	2.00	151	0:04:31
emotions	multi-label	8	1.08	303	6.60%	3.18	2.00	131	0:04:29
emotions	multi-label	8	1.16	313	11.18%	3.25	2.00	6	0:04:44
emotions	multi-label	8	1.32	310	18.06%	3.34	2.20	96	0:04:38
emotions	multi-label	16	-	256	0.00%	3.53	-	6	0:04:22
emotions	multi-label	16	1.02	259	3.09%	3.50	2.00	51	0:03:59
emotions	multi-label	16	1.04	268	4.10%	3.51	2.00	31	0:04:01
emotions	multi-label	16	1.08	265	5.28%	3.54	2.07	31	0:04:13
emotions	multi-label	16	1.16	279	9.32%	3.65	2.19	6	0:04:05
emotions	multi-label	16	1.32	279	13.26%	3.80	2.08	41	0:04:10
emotions	multi-label	32	-	240	0.00%	3.95	-	36	0:03:53
emotions	multi-label	32	1.02	236	1.27%	3.95	2.00	81	0:03:49
emotions	multi-label	32	1.04	239	1.26%	3.98	2.00	36	0:04:00
emotions	multi-label	32	1.08	239	2.09%	3.95	2.00	6	0:03:57
emotions	multi-label	32	1.16	247	4.05%	4.04	2.20	6	0:03:52
emotions	multi-label	32	1.32	269	9.29%	4.28	2.12	6	0:04:17
emotions	multi-label	64	-	195	0.00%	4.37	-	16	0:03:49
emotions	multi-label	64	1.02	194	1.03%	4.41	2.00	16	0:03:42
emotions	multi-label	64	1.04	204	1.96%	4.38	2.00	61	0:03:47
emotions	multi-label	64	1.08	194	3.09%	4.47	2.00	26	0:03:47
emotions	multi-label	64	1.16	207	3.38%	4.63	2.00	11	0:04:05
emotions	multi-label	64	1.32	266	6.02%	4.86	2.31	41	0:04:18
emotions	multi-label	128	-	218	0.00%	4.86	-	16	0:03:39
emotions	multi-label	128	1.02	215	0.47%	4.89	2.00	16	0:03:50
emotions	multi-label	128	1.04	227	0.44%	4.89	2.00	41	0:03:42
emotions	multi-label	128	1.08	206	1.94%	5.00	2.00	36	0:03:50
emotions	multi-label	128	1.16	204	3.43%	5.12	2.14	56	0:04:15
emotions	multi-label	128	1.32	239	8.79%	5.38	2.05	26	0:04:40
emotions	multi-label	256	-	184	0.00%	5.19	-	16	0:03:44
emotions	multi-label	256	1.02	204	0.49%	5.16	2.00	36	0:03:58
emotions	multi-label	256	1.04	183	1.09%	5.30	2.00	36	0:04:02

emotions	multi-label	256	1.08	173	2.89%	5.40	2.20	16	0:04:01
emotions	multi-label	256	1.16	214	2.80%	5.52	2.00	31	0:04:29
emotions	multi-label	256	1.32	253	11.07%	5.86	2.14	11	0:04:55
emotions	multi-label	512	-	178	0.00%	5.43	-	21	0:03:55
emotions	multi-label	512	1.02	183	1.09%	5.38	3.00	16	0:04:08
emotions	multi-label	512	1.04	179	0.00%	5.50	-	16	0:03:54
emotions	multi-label	512	1.08	187	1.07%	5.60	2.00	21	0:04:15
emotions	multi-label	512	1.16	210	1.90%	5.86	2.25	16	0:04:42
emotions	multi-label	512	1.32	234	6.41%	6.13	2.07	11	0:05:32

Table 19.: Model characteristics of multi-label models learned from the data set yeast

parameters				model characteristics					runtime
data set	head ref.	m	l	#rules	%MLHR	Avg.Cond.	Avg.MLH	opt. #rules	runtime
yeast	multi-label	0	-	6787	0.00%	1.85	-	6846	12:07:48
yeast	multi-label	0	1.02	3594	90.51%	1.88	2.00	3616	6:36:27
yeast	multi-label	0	1.04	3594	90.51%	1.88	2.00	3616	6:49:51
yeast	multi-label	0	1.08	3592	90.53%	1.88	2.00	3616	6:48:47
yeast	multi-label	0	1.16	3587	90.83%	1.88	2.00	3611	7:03:28
yeast	multi-label	0	1.32	3585	90.74%	1.88	2.00	3601	6:25:29
yeast	multi-label	1	-	4669	0.21%	2.55	3.00	4691	8:35:29
yeast	multi-label	1	1.02	3481	34.50%	2.52	2.00	3526	6:46:08
yeast	multi-label	1	1.04	3445	35.62%	2.52	2.00	3441	6:19:50
yeast	multi-label	1	1.08	3392	38.21%	2.55	2.00	3441	6:34:05
yeast	multi-label	1	1.16	3326	40.80%	2.59	2.01	3301	6:19:37
yeast	multi-label	1	1.32	3152	50.38%	2.68	2.02	3106	6:01:34
yeast	multi-label	2	-	4448	0.65%	2.66	3.00	4476	8:20:05
yeast	multi-label	2	1.02	3455	29.49%	2.59	2.00	3421	6:20:47
yeast	multi-label	2	1.04	3431	29.87%	2.59	2.00	3456	6:20:52
yeast	multi-label	2	1.08	3390	31.12%	2.62	2.00	3346	6:23:15
yeast	multi-label	2	1.16	3327	33.18%	2.67	2.01	3306	6:09:05
yeast	multi-label	2	1.32	3185	38.24%	2.75	2.11	3151	6:03:53
yeast	multi-label	4	-	4196	0.02%	2.78	3.00	4116	8:00:21
yeast	multi-label	4	1.02	3207	28.16%	2.73	2.00	3231	5:50:37
yeast	multi-label	4	1.04	3214	27.85%	2.73	2.00	3211	6:14:55
yeast	multi-label	4	1.08	3152	29.38%	2.80	2.01	3126	5:49:34
yeast	multi-label	4	1.16	3080	31.36%	2.87	2.07	3041	5:51:26
yeast	multi-label	4	1.32	2879	40.36%	3.03	2.34	2866	5:24:03
yeast	multi-label	8	-	3735	0.05%	3.01	3.00	3736	6:53:10
yeast	multi-label	8	1.02	2830	28.41%	2.94	2.00	2811	5:33:41
yeast	multi-label	8	1.04	2848	28.86%	2.96	2.00	2851	5:16:21
yeast	multi-label	8	1.08	2770	29.03%	3.04	2.02	2816	5:23:24
yeast	multi-label	8	1.16	2743	31.50%	3.17	2.15	2776	5:17:26
yeast	multi-label	8	1.32	2688	37.39%	3.35	2.32	2716	5:12:02
yeast	multi-label	16	-	3224	0.00%	3.32	-	3256	6:01:09
yeast	multi-label	16	1.02	2468	26.18%	3.29	2.00	2516	4:41:32
yeast	multi-label	16	1.04	2423	27.28%	3.35	2.01	2461	4:33:55

yeast	multi-label	16	1.08	2426	26.88%	3.43	2.07	2456	4:34:05
yeast	multi-label	16	1.16	2417	28.88%	3.58	2.15	2431	4:32:22
yeast	multi-label	16	1.32	2526	31.59%	3.75	2.30	2556	4:50:54
yeast	multi-label	32	-	2745	0.00%	3.77	-	2791	5:12:48
yeast	multi-label	32	1.02	2072	23.50%	3.70	2.00	6	4:09:44
yeast	multi-label	32	1.04	2083	23.62%	3.78	2.02	2086	4:00:49
yeast	multi-label	32	1.08	2169	22.96%	3.85	2.06	2171	4:09:18
yeast	multi-label	32	1.16	2225	23.91%	4.02	2.08	2231	4:23:26
yeast	multi-label	32	1.32	2313	29.10%	4.30	2.33	2336	4:32:40
yeast	multi-label	64	-	2371	0.00%	4.27	-	2326	4:45:58
yeast	multi-label	64	1.02	1823	19.58%	4.26	2.03	1876	3:33:46
yeast	multi-label	64	1.04	1833	19.15%	4.26	2.01	1856	3:46:31
yeast	multi-label	64	1.08	1929	20.11%	4.43	2.04	1946	3:42:22
yeast	multi-label	64	1.16	2051	20.04%	4.59	2.06	1996	4:03:44
yeast	multi-label	64	1.32	2191	27.80%	4.93	2.19	2171	4:18:49
yeast	multi-label	128	-	2050	0.00%	4.82	-	1971	4:15:59
yeast	multi-label	128	1.02	1605	17.69%	4.82	2.00	1616	3:15:42
yeast	multi-label	128	1.04	1597	17.09%	4.92	2.00	1611	3:30:18
yeast	multi-label	128	1.08	1725	17.10%	5.06	2.00	1731	3:32:28
yeast	multi-label	128	1.16	1877	18.86%	5.26	2.05	1871	3:59:59
yeast	multi-label	128	1.32	2031	25.85%	5.65	2.10	2101	4:21:23
yeast	multi-label	256	-	1883	0.00%	5.53	-	1786	4:10:09
yeast	multi-label	256	1.02	1453	15.76%	5.48	2.00	1506	3:13:10
yeast	multi-label	256	1.04	1528	15.77%	5.61	2.02	1521	3:24:02
yeast	multi-label	256	1.08	1627	18.25%	5.77	2.00	1621	3:35:47
yeast	multi-label	256	1.16	1701	18.34%	6.04	2.04	1766	3:55:25
yeast	multi-label	256	1.32	1988	24.85%	6.52	2.13	1921	4:34:22
yeast	multi-label	512	-	1683	0.00%	6.15	-	1666	3:58:35
yeast	multi-label	512	1.02	1418	14.67%	6.28	2.00	1396	3:24:46
yeast	multi-label	512	1.04	1444	15.37%	6.33	2.00	1486	3:38:48
yeast	multi-label	512	1.08	1569	14.91%	6.53	2.02	1571	3:54:31
yeast	multi-label	512	1.16	1690	19.35%	6.88	2.06	1726	4:20:30
yeast	multi-label	512	1.32	1926	25.55%	7.18	2.11	1966	4:51:28

Table 20.: Model characteristics of multi-label models learned from the data set medical

parameters				model characteristics					runtime
data set	head ref.	m	l	#rules	%MLHR	Avg.Cond.	Avg.MLH	opt. #rules	runtime
medical	multi-label	0	-	1562	0.00%	1.88	-	1631	0:04:47
medical	multi-label	0	1.02	1400	24.21%	1.96	2.00	1441	0:04:10
medical	multi-label	0	1.04	1400	24.21%	1.96	2.00	1441	0:04:11
medical	multi-label	0	1.08	1401	24.20%	1.97	2.00	1441	0:04:09
medical	multi-label	0	1.16	1401	24.27%	1.97	2.00	1441	0:04:16
medical	multi-label	0	1.32	1401	24.27%	1.97	2.00	1441	0:04:11
medical	multi-label	1	-	1129	0.00%	2.72	-	1201	0:03:47
medical	multi-label	1	1.02	1111	2.52%	2.73	2.00	1151	0:03:44
medical	multi-label	1	1.04	1109	2.71%	2.74	2.00	1171	0:03:39

medical	multi-label	1	1.08	1108	3.61%	2.74	2.00	1171	0:03:41
medical	multi-label	1	1.16	1107	5.51%	2.75	2.00	1161	0:03:43
medical	multi-label	1	1.32	1119	8.49%	2.76	2.01	1161	0:03:45
medical	multi-label	2	-	969	0.00%	2.93	-	1061	0:03:26
medical	multi-label	2	1.02	949	2.85%	2.93	2.00	1041	0:03:17
medical	multi-label	2	1.04	949	2.85%	2.94	2.00	1036	0:03:20
medical	multi-label	2	1.08	952	2.73%	2.95	2.00	1041	0:03:20
medical	multi-label	2	1.16	950	4.00%	2.94	2.00	1031	0:03:23
medical	multi-label	2	1.32	946	7.08%	2.98	2.01	1021	0:03:21
medical	multi-label	4	-	827	0.00%	3.16	-	896	0:02:57
medical	multi-label	4	1.02	814	2.46%	3.16	2.00	891	0:02:57
medical	multi-label	4	1.04	815	2.33%	3.16	2.00	891	0:02:54
medical	multi-label	4	1.08	817	2.08%	3.17	2.00	891	0:02:59
medical	multi-label	4	1.16	826	2.42%	3.18	2.00	896	0:02:54
medical	multi-label	4	1.32	823	13.24%	3.25	3.85	871	0:02:56
medical	multi-label	8	-	719	0.00%	3.46	-	746	0:02:37
medical	multi-label	8	1.02	714	1.54%	3.46	2.00	766	0:02:32
medical	multi-label	8	1.04	714	1.54%	3.46	2.00	766	0:02:35
medical	multi-label	8	1.08	717	1.39%	3.46	2.00	761	0:02:33
medical	multi-label	8	1.16	705	8.09%	3.55	3.37	746	0:02:34
medical	multi-label	8	1.32	708	12.85%	3.60	4.42	241	0:02:35
medical	multi-label	16	-	615	0.00%	3.77	-	86	0:02:16
medical	multi-label	16	1.02	619	1.78%	3.78	2.00	106	0:02:13
medical	multi-label	16	1.04	619	1.45%	3.79	2.00	106	0:02:14
medical	multi-label	16	1.08	617	6.32%	3.81	3.33	106	0:02:14
medical	multi-label	16	1.16	618	9.71%	3.88	3.95	106	0:02:18
medical	multi-label	16	1.32	614	14.98%	3.96	4.27	91	0:02:18
medical	multi-label	32	-	532	0.00%	4.12	-	151	0:02:04
medical	multi-label	32	1.02	543	1.10%	4.14	2.00	156	0:02:03
medical	multi-label	32	1.04	541	4.25%	4.15	3.22	156	0:02:02
medical	multi-label	32	1.08	543	6.81%	4.19	3.81	136	0:02:05
medical	multi-label	32	1.16	550	8.36%	4.30	4.07	136	0:02:06
medical	multi-label	32	1.32	544	13.97%	4.41	4.34	81	0:02:09
medical	multi-label	64	-	496	0.00%	4.38	-	86	0:01:54
medical	multi-label	64	1.02	491	3.87%	4.48	2.53	106	0:01:56
medical	multi-label	64	1.04	494	7.29%	4.46	3.64	106	0:01:58
medical	multi-label	64	1.08	501	8.18%	4.53	3.63	86	0:02:02
medical	multi-label	64	1.16	513	10.72%	4.61	4.24	131	0:02:02
medical	multi-label	64	1.32	504	16.27%	4.89	4.11	141	0:02:09
medical	multi-label	128	-	481	0.00%	4.53	-	121	0:01:58
medical	multi-label	128	1.02	476	6.09%	4.68	3.21	121	0:01:56
medical	multi-label	128	1.04	494	7.49%	4.72	3.54	121	0:02:00
medical	multi-label	128	1.08	496	10.08%	4.83	3.84	91	0:02:07
medical	multi-label	128	1.16	497	12.27%	5.01	3.82	41	0:02:10
medical	multi-label	128	1.32	504	17.66%	5.31	4.04	41	0:02:15
medical	multi-label	256	-	463	0.00%	4.65	-	56	0:01:56
medical	multi-label	256	1.02	465	6.45%	4.94	3.37	6	0:02:04

medical	multi-label	256	1.04	476	6.93%	4.99	3.24	6	0:02:05
medical	multi-label	256	1.08	474	9.49%	5.19	3.67	41	0:02:06
medical	multi-label	256	1.16	496	15.52%	5.32	3.84	41	0:02:19
medical	multi-label	256	1.32	491	20.16%	5.69	4.09	6	0:02:22
medical	multi-label	512	-	464	0.00%	4.77	-	6	0:01:59
medical	multi-label	512	1.02	461	5.42%	5.22	3.76	6	0:02:09
medical	multi-label	512	1.04	474	5.91%	5.25	3.68	6	0:02:16
medical	multi-label	512	1.08	484	10.12%	5.59	3.67	6	0:02:18
medical	multi-label	512	1.16	512	14.06%	5.68	4.03	6	0:02:25
medical	multi-label	512	1.32	520	18.85%	6.07	4.27	6	0:02:28

Table 21.: Model characteristics of multi-label models learned from the data set scene

parameters				model characteristics					runtime
data set	head ref.	m	l	#rules	%MLHR	Avg.Cond.	Avg.MLH	opt. #rules	runtime
scene	multi-label	0	-	2037	0.00%	1.97	-	656	7:51:43
scene	multi-label	0	1.02	1903	8.25%	1.97	2.00	6	7:27:49
scene	multi-label	0	1.04	1902	8.25%	1.98	2.00	6	7:12:52
scene	multi-label	0	1.08	1909	8.22%	1.97	2.00	6	7:37:39
scene	multi-label	0	1.16	1920	8.18%	1.98	2.00	6	7:19:41
scene	multi-label	0	1.32	1933	8.12%	1.98	2.00	6	7:44:53
scene	multi-label	1	-	1177	0.00%	2.77	-	361	4:34:38
scene	multi-label	1	1.02	1176	0.09%	2.77	2.00	361	4:20:25
scene	multi-label	1	1.04	1173	0.17%	2.76	2.00	361	4:43:11
scene	multi-label	1	1.08	1186	0.17%	2.76	2.00	361	4:27:54
scene	multi-label	1	1.16	1216	0.66%	2.75	2.00	16	4:58:06
scene	multi-label	1	1.32	1202	2.66%	2.73	2.00	6	4:49:34
scene	multi-label	2	-	1042	0.00%	2.93	-	11	3:49:57
scene	multi-label	2	1.02	1042	0.00%	2.93	-	11	4:03:06
scene	multi-label	2	1.04	1041	0.00%	2.93	-	11	4:08:03
scene	multi-label	2	1.08	1047	0.10%	2.95	2.00	11	3:59:51
scene	multi-label	2	1.16	1059	0.28%	2.98	2.00	11	4:24:05
scene	multi-label	2	1.32	1083	1.02%	3.00	2.00	11	4:12:58
scene	multi-label	4	-	893	0.00%	3.16	-	6	3:18:50
scene	multi-label	4	1.02	882	0.23%	3.16	2.00	6	3:18:45
scene	multi-label	4	1.04	876	0.23%	3.17	2.00	6	3:33:03
scene	multi-label	4	1.08	895	0.22%	3.17	2.00	6	3:31:08
scene	multi-label	4	1.16	897	0.11%	3.22	2.00	6	3:25:11
scene	multi-label	4	1.32	898	3.67%	3.29	2.73	6	3:47:30
scene	multi-label	8	-	734	0.00%	3.32	-	16	2:49:43
scene	multi-label	8	1.02	733	0.00%	3.32	-	16	2:43:58
scene	multi-label	8	1.04	725	0.14%	3.38	2.00	16	2:42:53
scene	multi-label	8	1.08	725	0.14%	3.43	2.00	16	2:52:45
scene	multi-label	8	1.16	717	0.70%	3.49	2.80	16	2:56:47
scene	multi-label	8	1.32	752	6.52%	3.52	2.96	16	3:06:22
scene	multi-label	16	-	573	0.00%	3.58	-	6	2:10:29
scene	multi-label	16	1.02	574	0.17%	3.59	2.00	6	2:17:19

scene	multi-label	16	1.04	562	0.00%	3.64	-	6	2:17:21
scene	multi-label	16	1.08	564	0.53%	3.72	2.00	6	2:14:11
scene	multi-label	16	1.16	594	3.70%	3.81	3.00	6	2:22:56
scene	multi-label	16	1.32	620	6.94%	3.94	2.98	6	2:40:54
scene	multi-label	32	-	464	0.00%	4.07	-	6	1:46:15
scene	multi-label	32	1.02	465	0.00%	4.07	-	6	1:52:15
scene	multi-label	32	1.04	452	0.00%	4.12	-	6	1:46:56
scene	multi-label	32	1.08	429	1.17%	4.20	3.00	6	1:52:35
scene	multi-label	32	1.16	487	3.29%	4.31	3.00	6	2:05:11
scene	multi-label	32	1.32	554	4.69%	4.47	3.00	6	2:11:44
scene	multi-label	64	-	341	0.00%	4.49	-	6	1:30:00
scene	multi-label	64	1.02	356	0.28%	4.53	2.00	6	1:33:13
scene	multi-label	64	1.04	375	0.80%	4.54	2.33	6	1:31:40
scene	multi-label	64	1.08	389	2.06%	4.62	2.88	6	1:41:52
scene	multi-label	64	1.16	407	2.70%	4.83	3.00	6	1:50:19
scene	multi-label	64	1.32	492	3.05%	5.15	3.00	6	2:13:06
scene	multi-label	128	-	290	0.00%	5.04	-	6	1:15:13
scene	multi-label	128	1.02	300	0.33%	5.09	2.00	6	1:19:18
scene	multi-label	128	1.04	308	0.65%	5.18	2.50	6	1:21:19
scene	multi-label	128	1.08	333	0.90%	5.24	2.67	6	1:31:57
scene	multi-label	128	1.16	357	2.24%	5.50	2.88	6	1:43:39
scene	multi-label	128	1.32	414	3.86%	5.83	3.00	6	2:12:54
scene	multi-label	256	-	257	0.00%	5.52	-	6	1:13:45
scene	multi-label	256	1.02	264	0.38%	5.57	3.00	6	1:16:32
scene	multi-label	256	1.04	285	0.00%	5.60	-	6	1:23:22
scene	multi-label	256	1.08	310	0.32%	5.93	3.00	6	1:34:36
scene	multi-label	256	1.16	331	1.21%	6.18	3.00	6	1:41:55
scene	multi-label	256	1.32	421	4.51%	6.62	2.79	6	2:17:52
scene	multi-label	512	-	236	0.00%	5.94	-	6	1:16:37
scene	multi-label	512	1.02	244	0.00%	5.90	-	6	1:20:01
scene	multi-label	512	1.04	256	0.39%	6.29	3.00	6	1:29:06
scene	multi-label	512	1.08	302	0.66%	6.47	2.50	6	1:36:32
scene	multi-label	512	1.16	362	0.28%	6.78	3.00	6	1:58:57
scene	multi-label	512	1.32	441	3.85%	7.29	2.82	6	2:29:19

Table 22.: Model characteristics of single-label models learned from the data set birds

parameters			model characteristics		runtime
data set	head ref.	m	#rules	Avg.Cond.	runtime
birds	single-label	0	518	1.53	0:06:30
birds	single-label	1	446	2.06	0:05:55
birds	single-label	2	421	2.17	0:05:37
birds	single-label	4	393	2.35	0:05:15
birds	single-label	8	355	2.54	0:04:51
birds	single-label	16	324	2.77	0:04:27
birds	single-label	32	308	3.01	0:04:14
birds	single-label	64	288	3.39	0:04:03

birds	single-label	128	249	3.89	0:03:58
birds	single-label	256	247	4.47	0:04:04
birds	single-label	512	237	4.67	0:04:12

Table 23.: Model characteristics of single-label models learned from the data set emotions

parameters			model characteristics		runtime
data set	head ref.	m	#rules	Avg.Cond.	runtime
emotions	single-label	0	841	1.74	0:03:28
emotions	single-label	1	509	2.58	0:02:13
emotions	single-label	2	439	2.73	0:01:58
emotions	single-label	4	399	2.88	0:01:44
emotions	single-label	8	334	3.12	0:01:29
emotions	single-label	16	256	3.53	0:01:19
emotions	single-label	32	240	3.95	0:01:14
emotions	single-label	64	195	4.37	0:01:08
emotions	single-label	128	218	4.86	0:01:09
emotions	single-label	256	184	5.19	0:01:11
emotions	single-label	512	178	5.43	0:01:13

Table 24.: Model characteristics of single-label models learned from the data set yeast

parameters			model characteristics		runtime
data set	head ref.	m	#rules	Avg.Cond.	runtime
yeast	single-label	0	6787	1.85	3:12:55
yeast	single-label	1	4689	2.55	2:19:52
yeast	single-label	2	4506	2.65	2:13:54
yeast	single-label	4	4198	2.78	2:04:54
yeast	single-label	8	3739	3.01	1:52:08
yeast	single-label	16	3224	3.32	1:37:50
yeast	single-label	32	2745	3.77	1:24:19
yeast	single-label	64	2371	4.27	1:12:55
yeast	single-label	128	2050	4.82	1:05:49
yeast	single-label	256	1883	5.53	1:03:31
yeast	single-label	512	1683	6.15	1:03:46

Table 25.: Model characteristics of single-label models learned from the data set medical

parameters			model characteristics		runtime
data set	head ref.	m	#rules	Avg.Cond.	runtime
medical	single-label	0	1562	1.88	0:01:46
medical	single-label	1	1129	2.72	0:01:25
medical	single-label	2	969	2.93	0:01:17
medical	single-label	4	827	3.16	0:01:07
medical	single-label	8	719	3.46	0:00:59
medical	single-label	16	615	3.77	0:00:51

medical	single-label	32	532	4.12	0:00:47
medical	single-label	64	496	4.38	0:00:44
medical	single-label	128	481	4.53	0:00:43
medical	single-label	256	463	4.65	0:00:44
medical	single-label	512	464	4.77	0:00:44

Table 26.: Model characteristics of single-label models learned from the data set scene

parameters			model characteristics		runtime
data set	head ref.	m	#rules	Avg.Cond.	runtime
scene	single-label	0	2037	1.97	2:20:03
scene	single-label	1	1177	2.77	1:23:28
scene	single-label	2	1042	2.93	1:13:04
scene	single-label	4	893	3.16	1:02:37
scene	single-label	8	734	3.32	0:51:36
scene	single-label	16	573	3.58	0:40:57
scene	single-label	32	464	4.07	0:33:21
scene	single-label	64	341	4.49	0:28:16
scene	single-label	128	290	5.04	0:23:59
scene	single-label	256	257	5.52	0:23:16
scene	single-label	512	236	5.94	0:23:06

---

## Bibliography

---

- Alpaydin, Ethem (2010). *Introduction to Machine Learning*. 2nd. The MIT Press.
- Chen, Tianqi and Carlos Guestrin (Aug. 2016). “XGBoost”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Dembczynski, Krzysztof et al. (2012). “On label dependence and loss minimization in multi-label classification”. In: *Mach. Learn.* 88(1-2), pp. 5–45.
- Fürnkranz, Johannes (1999). “Separate-and-conquer rule learning”. In: *Artificial Intelligence Review* 13, pp. 3–54.
- Fürnkranz, Johannes et al. (Nov. 2008). “Multilabel classification via calibrated label ranking”. In: *Machine Learning* 73, pp. 133–153.
- Hefter, Pascal Dominik (Jan. 2020). “Learning Multi-Label Rules Using Bottom-Up Rule Induction”. Bachelor thesis. Knowledge Engineering Group, TU Darmstadt.
- Janssen, Frederik (Oct. 2012). “Heuristic Rule Learning”. PhD thesis. Darmstadt: Technische Universität.
- Johannes, Fillemon and Gerhard Widmer (Jan. 1996). “Incremental Reduced Error Pruning”. In: Klein, Yannik (Oct. 2018). *Discovery of Multilabel Rules by Relaxed Pruning*. Bachelor thesis.
- Koyejo, Oluwasanmi O et al. (2015). “Consistent Multilabel Classification”. In: *Advances in Neural Information Processing Systems* 28. Ed. by C. Cortes et al. Curran Associates, Inc., pp. 3321–3329.
- Loza Mencía, Eneldo (2012). “Efficient Pairwise Multilabel Classification”. PhD thesis. Darmstadt: Technische Universität.
- Loza Mencía, Eneldo and Frederik Janssen (2016). “Learning rules for multi-label classification: a stacking and a separate-and-conquer approach”. In: *Machine Learning* 105(1).
- Mehta, Manish et al. (1996). “SLIQ: A fast scalable classifier for data mining”. In: *Advances in Database Technology — EDBT ’96*. Ed. by Peter Apers et al. Springer Berlin Heidelberg: Berlin, Heidelberg, pp. 18–32.
- Mohri, Mehryar et al. (2018). *Foundations of Machine Learning*. 2nd ed. Adaptive Computation and Machine Learning. MIT Press: Cambridge, MA. 504 pp.
- Rapp, Michael (Dec. 2016). “A Separate-and-Conquer Algorithm for Learning Multi-Label Head Rules”. Master thesis. TU Darmstadt, Knowledge Engineering Group.
- Rapp, Michael, Eneldo Loza Mencía, et al. (2018). “Exploiting Anti-monotonicity of Multi-label Evaluation Measures for Inducing Multi-label Rules”. In: *CoRR* abs/1812.06833.
- Rapp, Michael, Eneldo Loza Mencía, et al. (2019). *On the Trade-off Between Consistency and Coverage in Multi-label Rule Learning Heuristics*.
- Rapp, Michael, Eneldo Loza Mencía, et al. (2020). *Learning Gradient Boosted Multi-label Classification Rules*.
- Read, Jesse et al. (2009). “Classifier Chains for Multi-label Classification”. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Wray Buntine et al. Springer Berlin Heidelberg: Berlin, Heidelberg, pp. 254–269.
- Shafer, John et al. (Aug. 2000). “SPRINT: A scalable parallel classifier for data mining”. In: *VLDB*.
- Spolaôr, Newton et al. (Mar. 2013). “A Comparison of Multi-label Feature Selection Methods using the Problem Transformation Approach”. In: *Electronic Notes in Theoretical Computer Science* 292, pp. 135–151.

- 
- Tsamardinos, Ioannis et al. (2017). “Bootstrapping the Out-of-sample Predictions for Efficient and Accurate Cross-Validation”. In: *CoRR* abs/1708.07180.
- Tsoumakas, Grigorios et al. (July 2011). “MULAN: A Java library for multi-label learning”. In: *Journal of Machine Learning Research* 12, pp. 2411–2414.
- Varma, Sudhir and Richard Simon (Feb. 2006). “Bias in Error Estimation When Using Cross-Validation for Model Selection.” *BMC Bioinformatics*, 7(1), 91”. In: *BMC bioinformatics* 7, p. 91.
- Zhang, Min-Ling, Yu-Kun Li, et al. (Nov. 2017). “Binary relevance for multi-label learning: an overview”. In: *Frontiers of Computer Science* 12.
- Zhang, Min-Ling and Kun Zhang (2010). “Multi-Label Learning by Exploiting Label Dependency”. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’10. Association for Computing Machinery: Washington, DC, USA, pp. 999–1008.

---

## **Erklärung zur Abschlussarbeit gemäß §22 Abs. 7 APB der TU Darmstadt**

---

Hiermit versichere ich, Jakob Steeg, die vorliegende Bachelorarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 06.10.2020

---

J. Steeg