Multi-Directional Rule Set Learning

Jonas Schouterden[1,2][0000−0003−3071−7259], Jesse Davis[1,2][0000−0002−3748−9263], and Hendrik Blockeel[1,2][0000−0003−0378−3699]

[1] KU Leuven, Department of Computer Science
Celestijnenlaan 200A box 2402, 3001 Leuven, Belgium
[2] Leuven.AI
{jonas.schouterden,jesse.davis,hendrik.blockeel}@cs.kuleuven.be

Abstract. A rule set is a type of classifier that, given attributes X, predicts a target Y. Its main advantage over other types of classifiers is its simplicity and interpretability. A practical challenge is that the end user of a rule set does not always know in advance which target will need to be predicted. One way to deal with this is to learn a multi-directional rule set, which can predict any attribute from all others. An individual rule in such a multi-directional rule set can have multiple targets in its head, and thus be used to predict any one of these. Compared to the naive approach of learning one rule set for each possible target and merging them, a multi-directional rule set containing multi-target rules is potentially smaller and more interpretable. Training a multi-directional rule set involves two key steps: generating candidate rules and selecting rules. However, the best way to tackle these steps remains an open question. In this paper, we investigate the effect of using Random Forests as candidate rule generators and propose two new approaches for selecting rules with multi-target heads: MIDS, a generalization of the recent single-target IDS approach, and RR, a new simple algorithm focusing only on predictive performance. Our experiments indicate that (1) using multi-target rules leads to smaller rule sets with a similar predictive performance, (2) using Forest-derived rules instead of association rules leads to rule sets of similar quality, and (3) RR outperforms MIDS, underlining the usefulness of simple selection objectives.

Keywords: Rule learning· Multi-directional models· Association rule mining· Decision trees

1 Introduction

Rule sets are classifiers predicting one target Y given attributes X. Their popularity stems from their simplicity and interpretability. A problem in practice is that a rule set's user might not know during training which attribute needs to be predicted. Examples of such cases are missing value imputation, where there are gaps in the data, or anomaly detection, where a value of a suspicious instance might be compared with a value representative of the training data. In such cases, the user would need to learn a separate rule set for each attribute.

Learning one rule set per attribute negatively impacts the collective interpretability, as the bodies of rules predicting correlated targets cannot be shared. If rules could predict multiple targets, the rule sets (1) might be more interpretable by using fewer rules (as a single rule can predict multiple targets at once (Sec. 3.2)), and (2) might explicate correlations between different targets.

While using multi-target rules might help, current rule set algorithms selecting a subset of rules $\mathcal{R}_{sel}$ out of a candidate rule set $\mathcal{R}_{cand}$ only work with single-target rules. To work with multi-target rules, they would need to simultaneously optimize the predictive performance for multiple targets.

Another problem is that as the candidate rule set $\mathcal{R}_{cand}$ typically consists of association rules, the user must set a confidence and support threshold in advance without knowing what the size or quality of the resulting rule set will be. Too low thresholds cause $\mathcal{R}_{cand}$ to become too large, potentially making both the rule set generation and rule set selection intractable. Too high thresholds may result in a small $\mathcal{R}_{cand}$ limiting the number of rules that can be selected, which might result in an selected subset $\mathcal{R}_{sel}$ of lesser quality. As association rule mining is often very sensitive to these thresholds, a small change in value might lead to candidate sets of widely varying sizes.

In summary, current rule set methods based on selecting a subset of candidate rules have the following problems: (1) they require the user to specify the target in advance, (2) they cannot select multi-target rules, and (3) they often use association rules, which are difficult to control in number and quality.

To address these problems, this paper investigates how to learn a *multi-directional rule set* able to predict any attribute given all other attributes, thus no longer requiring the user to specify the target in advance. We propose two multi-target rule selection approaches: a generalization of *Interpretable Decision Sets (IDS)* [11], and *RR*, a new algorithm focusing only on selecting a rule set with a high predictive performance for all targets. Finally, we propose to derive rule sets from Random Forests, as the number and size of trees in a Random Forest is easy to control, and they are learned to do prediction. Our experiments indicate that (1) using multi-target rules leads to smaller rule sets with a similar predictive performance, (2) using tree rules instead of association rules leads to rule sets of similar quality, and (3) RR outperforms MIDS, underlining the usefulness of simple selection objectives.

The rest of this paper is structured as follows. After Section 2 provides references to related work, Sections 3 and 4 introduce the predictive settings, rule (set) representations and rule generation approaches used in this paper. Sections 5 and 6 describe RR and MIDS. An experimental evaluation is provided in Section 7, after which Section 8 gives a conclusion.

2 Related work

Rule learning [5] can be divided into (1) predictive approaches for building classifiers, and (2) descriptive approaches for discover interesting patterns in data in the form of rules. These two groups are bridged by the LeGo framework [6],

of which associative classifiers [12,2] are a prototypical instantiation. Associative classifiers are typically learned in three stages [6]. First, a set of candidate rules $\mathcal{R}_{cand}$ is mined from data [1]. Second, a subset of those rules $\mathcal{R}_{sel} \subseteq \mathcal{R}_{cand}$ is selected which optimizes some rule set objective. Third, the selected rules are combined to form a classifier. Different candidate rule generation and rule selection approaches can be combined, as they are often independent. CBA [12] is one of the oldest and best-known associative classifiers, selecting association rules based on their confidence. In this paper, we propose two multi-directional associative classifiers: MIDS and RR. MIDS generalizes the recent IDS [11] to support multi-target rules. RR is a a new algorithm. However, other multi-target classifiers exist [15]. Predictive clustering rules [16] is a coverage-based multi-target rule learning approach keeping a clear separation between descriptive and target attributes. Other examples are PGMs [10], which use a graph structure instead of logical rules, and MERCS [14] models, which use decision trees.

3 Preliminaries

In this section, we first introduce the single-target and multi-directional prediction settings used in this paper. Second, we define the representation of rules and rule sets. Third, we point out the necessity of tie-breaking strategies and default predictions in associative classification.

3.1 Predictive settings

In the single-target setting, a learned model predicts a designated target attribute Y from m descriptive attributes $X_j \in \boldsymbol{X}$. Here, the training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ contains N attribute-value examples. In a multi-directional setting, the target is not known in advance: the learned model must be able to predict any attribute given all other attributes. Here, the training set $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ has m attributes X_j and no distinction is made between descriptive and target attributes. The value of attribute X_j for datapoint $\boldsymbol{x}$ is $\boldsymbol{x}[X_j]$.

3.2 Rule set representations

This paper considers rules of the form:

$$r = \mathit{body}(r) \rightarrow \mathit{head}(r) = b_1 \wedge \cdots \wedge b_{r_b} \rightarrow h_1 \wedge \cdots \wedge h_{r_h}$$

where each h_i and b_i is of the form $(X_j, \mathit{operator}, \mathit{value})$. Abusing notation, $\mathit{head}(r)$ and $\mathit{body}(r)$ denote both the set and conjunction of those literals, and $\mathit{length}(r) = |\mathit{head}(r)| + |\mathit{body}(r)|$ denotes the length of rule r. Using attr to denote the attributes in a head or body, $\mathit{attr}(\mathit{head}(r)) \cap \mathit{attr}(\mathit{body}(r)) = \emptyset$, and both $\mathit{head}(r)$ and $\mathit{body}(r)$ are not empty. A rule is single-target if $|\mathit{head}(r)| = 1$ and otherwise it is multi-target. All literals in the *head* use equality as the *operator*.

A single-target rule set consist of only single-target rules, but a multi-directional rule set may consist of either single-target rules (with different rules predicting different targets) or multi-target rules (where a single rule can predict multiple targets at once).

3.3 Tie breaking functions and default predictions

As the rules of an associative classifier might overlap, an instance might be covered by multiple rules. As a result, a tie-breaking strategy is necessary to get a single prediction. Different strategies exist, such as (weighted) voting, or only using the rule with the highest F1-score. Also, as the rules might not cover the whole instance space, a default prediction is necessary when no rule applies. A common choice is the attribute's mode in the training data.

4 Rule generation

In this paper, a candidate rule set is generated with either association rule mining or decision tree ensembles. To mine association rules, each example $\boldsymbol{x} \in \mathcal{D}$ is transformed into a transaction containing m items of the form '$X_j = v_j$' with v_j in the domain of X_j ($v_j \in dom(X_j)$), on which frequent itemset mining can be used. As this requires categorical attributes, numerical attributes are discretized.

To derive a rule set from a tree ensemble, each tree is converted into its corresponding rule set [13]. Each rule corresponds to a path in a decision tree from the root to a leaf node. The rule's body consists of the tests in the inner nodes, while its head consists of the predictions in the leaf node.

5 RR: a simple multi-target rule selection approach

This section proposes *RR*, a new algorithm that greedily selects multi-target rules from a candidate rule set. *RR* is purely based on maximizing the predictive performance of the resulting classifier.

Algorithm 1 outlines *RR*, which iteratively selects a rule increasing the F1-score of one target while limiting a possible score decrease on the other targets. Its input is a multi-target candidate rule set $\mathcal{R}_{cand}$. It starts with an empty set initial classifier $\mathcal{R}_{sel}$. *RR* adds rules to $\mathcal{R}_{sel}$ by selecting one rule at a time from $\mathcal{R}_{cand} \setminus \mathcal{R}_{sel}$. To select rules, the algorithm loops over the target attributes in a round-robin fashion (thus the name *RR*), focusing on each target in turn. When focusing on a target attribute X_j, *RR* must select a rule r_{sel} that increases the F1-score of the complete rule set $\mathcal{R}_{sel}$ for X_j. However, selecting a multi-target rule changes the F1-scores for all targets in the head of that rule. That is, adding a rule increasing the F1-score of the current target X_j might decrease the F1-scores of other targets $X_o \neq X_j$. To deal with this, *RR* first finds the rule r_{best} that if added to $\mathcal{R}_{sel}$ results in the largest F1-score increase for the current target X_j. Second, it finds all rules $\mathcal{R}_{X_j,\delta}$ that, when individually added to $\mathcal{R}_{sel}$, result in a F1-score that differs by at most δ from the F1-score for $\mathcal{R}_{sel} \cup \{r_{best}\}$ when predicting the current target X_j. Third, *RR* selects the rule from $\mathcal{R}_{X_j,\delta}$ that decreases the F1-scores on the other targets the least. That is, δ allows trading off selecting the better rule for the current target with the 'damage' done to other targets. *RR* stops if no rule can be found that increases the F1-score of $\mathcal{R}_{sel}$ on any target X_t by at least ϵ, to prevent overfitting. At the end, $\mathcal{R}_{sel}$ contains the rules to be used as classifier.

Algorithm 1 Round-Robin (RR). Note: $\mathcal{R} + r$ is short for $\mathcal{R} \cup \{r\}$.

Require:
$\mathcal{R}_{cand}$, the candidate rule set.
ϵ, the minimally required increase in F1-score when adding a rule.
δ, the maximum distance a rule can be to the best rule to be considered.
$score_i(\mathcal{R})$, the F1-score of rule set $\mathcal{R}$ for attribute X_i on the training data.

procedure ROUND_ROBIN
  $\mathcal{R}_{sel} \leftarrow \emptyset$
  while $\exists$ target $X_t : select\_rule\_for(X_t, \mathcal{R}_{sel}) \neq None$ **do**
    $X_j \leftarrow$ the next target in a round-robin fashion.
    $r_{sel} \leftarrow select\_rule\_for(X_j, \mathcal{R}_{sel})$
    if $r_{sel} \neq None$ **then**
      $\mathcal{R}_{sel} \leftarrow \mathcal{R}_{sel} + r_{sel}$
  return $\mathcal{R}_{sel}$
procedure SELECT_RULE_FOR(target X_j, $\mathcal{R}_{sel}$)
  $\mathcal{R}_{X_j} \leftarrow \{r \in \mathcal{R}_{cand} \setminus \mathcal{R}_{sel} \mid X_j \in head(r) \wedge score_j(\mathcal{R}_{sel} + r) - score_j(\mathcal{R}_{sel}) > \epsilon\}$
  if $\mathcal{R}_{X_j} == \emptyset$ **then**
    return *None*
  else
    $r_{best} \leftarrow \arg\max_{r \in R_{X_j}} score_j(\mathcal{R}_{sel} + r)$
    $\mathcal{R}_{X_j,\delta} \leftarrow \{r \in \mathcal{R}_{X_j} \mid score_j(\mathcal{R}_{sel} + r_{best}) - score_j(\mathcal{R}_{sel} + r) < \delta\}$
    $r_{sel} \leftarrow \arg\max_{r \in \mathcal{R}_{X_j,\delta}} \big(\min_{X_o \in head(r) \setminus X_j} score_o(\mathcal{R}_{sel} + r) - score_o(\mathcal{R}_{sel}) \big) \big)$
    return r_{sel}

6 MIDS: Multi-target IDS

As a second multi-target rule selection approach, we propose *Multi-target Interpretable Decision Sets (MIDS)*, a generalization of *Interpretable Decision Sets (IDS)* [11]. We choose IDS as it is a recent single-target approach offering a high predictive performance and interpretability with a small rule set size. Section 6.1 introduces IDS on a high level. In Section 6.2, we generalize the IDS objective function to support multi-target rules.

6.1 IDS: (Single-Target) Interpretable Decision Sets

First, IDS specifies to generate a candidate association rule set $\mathcal{R}_{cand}$ using Apriori, which we substitute for the more efficient FP-growth [7]. Second, IDS selects a subset $\mathcal{R}_{sel} \subseteq \mathcal{R}_{cand}$ that (locally) maximizes an objective function $f(\mathcal{R})$. The objective function is a weighted sum of several heuristics f_i indicating the rule set quality, such as the predictive performance, the size and the interpretability of the rule set:

$$\mathcal{R}_{sel} = \arg\max_{\mathcal{R} \subseteq \mathcal{R}_{cand}} f(\mathcal{R}) = \arg\max_{\mathcal{R} \subseteq \mathcal{R}_{cand}} \sum_{i=1}^{7} \lambda_i f_i(\mathcal{R}) \tag{1}$$

Section 6.2 explains the different sub-objectives f_i and how we generalize them to support multi-target rules.

The final subset $\mathcal{R}_{sel}$ is used for classification. IDS suggests using the rule with the highest F1-score as a tie-breaking strategy and to predict the majority class label in the training data as default prediction. However, the user is free to choose other tie-breaking and default prediction strategies.

Unconstrained submodular maximization Finding the best subset $\mathcal{R}_{sel} \subseteq \mathcal{R}_{cand}$ that maximizes some objective function f (Eq. 1) corresponds to a combinatorial optimization problem. By formulating f as a non-negative non-normal unconstrained submodular maximization problem, a general algorithm for this problem type can be used for IDS. However, as maximizing an unconstrained submodular function is NP-hard [4], polynomial algorithms only guarantee to find a local optimum. Originally, IDS [11] specified to use the Smooth Local Search (SLS) algorithm [4]. However, as using SLS with IDS can be prohibitively slow [8], we choose to use the more recent Randomized Double Greedy Search algorithm [3], which is considerably faster and has better theoretical guarantees.

6.2 From IDS to MIDS: adding support for multi-target rules

To select a rule set $\mathcal{R}_{sel} \subseteq \mathcal{R}_{cand}$, IDS maximizes an objective function composed of 7 sub-objectives (see Eq. 1). The sub-objectives can be loosely divided into four groups, quantifying different aspects of a rule set $\mathcal{R}$. The first group focuses on rule set conciseness, the second on non-overlapping decision boundaries, the third on explaining as many attribute-values as possible, while the fourth group focuses on making accurate predictions. To trade off the importance of each of these aspects, the original work suggests that the weights λ_i can either be set by the user or be found using coordinate ascent.

Next, we modify the IDS sub-objectives in two ways[3]: we add (1) normalization and (2) support for multi-target rules. First, we normalize each of the sub-objectives to be in the interval $[0, 1]$. While the sub-objectives of the original IDS are non-negative, they do not have a clear upper bound. This might result in some sub-objectives dominating over others, but it also makes it difficult for the user to choose weights λ_i. When compared to the original specification, our normalization corresponds to multiplying the weight λ_i with a constant dependent on the candidate rule set $\mathcal{R}_{cand}$.

Second, we modify the IDS sub-objectives to support multi-target rules. Our generalization collapses to the original formulation when using single-target rules. To stay close to the original IDS specification, we do not further modify the sub-objectives, but indicate possible improvements as footnotes.

[3] Note that while IDS specifies it uses association rules, no modifications are necessary to support rules derived from decision trees. Any rule type for which the coverage and overlap can be calculated is supported.

Rule set conciseness The first two sub-objectives f_1 and f_2 directly correspond to those of IDS, apart from the normalization. The first minimizes the number of rules selected from the candidate rule set, while the second minimizes the total number of literals in the rule set[4]:

$$f_1(\mathcal{R}) = 1 - \frac{|\mathcal{R}|}{|\mathcal{R}_{cand}|}$$
$$f_2(\mathcal{R}) = 1 - \frac{1}{L_{max} \cdot |\mathcal{R}_{cand}|} \sum_{r \in \mathcal{R}} \mathit{length}(r)$$
$$L_{max} = \max_{r \in \mathcal{R}_{cand}} \mathit{length}(r)$$

Non-overlapping decision boundaries Two IDS sub-objectives f_3 and f_4 minimize the overlap of rules predicting a value for its target attribute Y. IDS assumes that a rule set with lower overlap is easier interpret, as fewer rules make predictions for a given example. While rule overlap in IDS is implicitly relative to its single target, we generalize the definition of rule overlap to be relative to a given target attribute X_j.

We define two rules r_1, r_2 to overlap relative to an attribute X_j if they share a covered example and both predict a value for attribute X_j:

$$\mathit{cover}(r) = \{\boldsymbol{x} \in \mathcal{D} \mid \boldsymbol{x} \models \mathit{body}(r)\}$$
$$\mathit{overlap}_j(r_1, r_2) = \begin{cases} \mathit{cover}(r_1) \cap \mathit{cover}(r_2) & \text{if } X_j \in \mathit{attr}(\mathit{head}(r_1)) \cap \mathit{attr}(\mathit{head}(r_2)) \\ \emptyset & \text{if } X_j \notin \mathit{attr}(\mathit{head}(r_1)) \cap \mathit{attr}(\mathit{head}(r_2)) \end{cases}$$

The goal of f_3 is to minimize the overlap of rules predicting the same value for a given target attribute. To generalize this to the multi-target case, we average over the m different targets, normalizing each contribution. Following the original IDS, $N \cdot |R_{cand,X_j}|^2$ is used as a simple upper bound for the maximal overlap relative to an attribute X_j given a training set of N instances[5]:

$$f_3(\mathcal{R}) = \frac{1}{m} \sum_{j=1}^{m} \Big[1 - \frac{1}{N \cdot |R_{cand,X_j}|^2} \sum_{\substack{r_k, r_l \in \mathcal{R} \\ k<l \\ (X_j = c_k) \in \mathit{head}(r_k) \\ (X_j = c_l) \in \mathit{head}(r_l) \\ c_k = c_l}} |\mathit{overlap}_j(r_k, r_l)|\Big]$$
$$R_{cand,X_j} = \{r \in \mathcal{R}_{cand} \mid X_j \in \mathit{attr}(\mathit{head}(r))\}$$

Sub-objective f_4 minimizes the overlap of rules predicting a *different* value for a given target attribute, which corresponds to substituting $c_k = c_l$ by $c_k \neq c_l$ when filtering the sum in the formulation of f_3 above.

[4] A better denominator is to use $\sum_{r \in \mathcal{R}_{cand}} \mathit{length}(r)$ instead of $L_{max} \cdot |\mathcal{R}_{cand}|$.
[5] A stricter upper bound is $\frac{N}{2} \cdot |R_{cand,X_j}| \cdot (|R_{cand,X_j}| - 1)$.

Predicting all attribute-values IDS sub-objective f_5 formulates the assumption that a user wants for each value c in a target's domain $dom(Y)$ at least one rule that explains it. We generalize this for multi-target rules by averaging the normalized contributions for each different target:

$$f_5(\mathcal{R}) = \frac{1}{m}\sum_{j=1}^{m}\frac{1}{|dom(X_j)|}\sum_{c' \in dom(X_j)} \mathbb{1}\Big[\exists r \in \mathcal{R} \mid (X_j = c') \in head(r)\Big]$$

Predictive performance Two sub-objectives focus on the predictive performance of the rule set. To generalize these objectives to a multi-directional setting, we first define the (in)correct coverage of a rule as the set of (in)correctly classified examples relative to a given target attribute:

$$\textit{correct-cover}_j(r) = \{\boldsymbol{x} \in \textit{cover}(r) \mid (X_j = c_j) \in \textit{head}(r) \text{ and } \boldsymbol{x}[X_j] = c_j\}$$
$$\textit{incorrect-cover}_j(r) = \{\boldsymbol{x} \in \textit{cover}(r) \mid (X_j = c_j) \in \textit{head}(r) \text{ and } \boldsymbol{x}[X_j] \neq c_j\}$$

Sub-objective f_6 prefers rules predicting few examples incorrectly. Its generalization to the multi-directional setting averages the number of mistakes each rule makes over that rule's target attributes:

$$f_6(\mathcal{R}) = 1 - \frac{1}{N \cdot |\mathcal{R}_{cand}|}\sum_{r \in \mathcal{R}} \textit{avg-incorrect-cover-size}(r)$$
$$\textit{avg-incorrect-cover-size}(r) = \frac{1}{|attr(head(r))|}\sum_{X_j \in attr(head(r))} |\textit{incorrect-cover}_j(r)|$$

Sub-objective f_7 focuses on each attribute-value of each instance being correctly predicted by at least one rule:

$$f_7(\mathcal{R}) = \frac{1}{N \cdot m}\sum_{\boldsymbol{x} \in \mathcal{D}}\sum_{j=1}^{m} \mathbb{1}\Big[|\{r \in \mathcal{R} \mid \boldsymbol{x} \in \textit{correct-cover}_j(r)\}| \geq 1\Big]$$

7 Experimental evaluation

In this section, we use two questions to experimentally investigate whether our proposed rule generation and rule selection approaches can lead to better rule models than using single-target association rules.

First, **(Q1)** "do tree rules lead to better models than association rules?" To answer this, we compare association rules with tree rules in a single-target prediction setting using IDS as the single-target rule selection approach.

Second, **(Q2)** "does learning a multi-directional model from multi-target rules have advantages over using a collection of single-target rule models?" To answer this, we compare single-target and multi-target tree-derived rules in a multi-directional prediction setting using three rule selection methods: IDS for the single-target rules, RR and MIDS for the multi-target rules.

Our Python code is available on GitHub.[6]

[6] `https://github.com/joschout/Multi-Directional-Rule-Set-Learning`

7.1 General methodology

We use 7 UCI datasets, provided in discretized form by the arcBench benchmarking suite [9]: iris, diabetes, glass, segment, breast-w and vehicle. The discretization is required for association rule mining. We learn and evaluate all models using 10-fold cross validation. When comparing two models, we use a Wilcoxon signed-rank test with a significance level $\alpha = 0.05$.

We use the same tie-breaking and default prediction strategies for all models. As the tie-breaking strategy, we use weighted voting, where each rule gets a vote weighted by the rule's confidence in the training set. As a default prediction for each target attribute, we use its majority value in the training set.

For (M)IDS, we use as optimization algorithm 'Double Greedy Local Search' (unlike the original IDS; see Section 6.1). Since the optimization uses randomization to find a *locally* optimal rule set $\mathcal{R}_{sel}$, we run each (M)IDS configuration 10 times and pick the rule set with the highest objective function value. For both IDS and MIDS, we use the same implementation including normalization of the sub-objectives with all weights set to $\lambda_i = 1$.

Compared metrics We investigate the predictive performance, model induction time, model size and interpretability of the selected rule models $\mathcal{R}_{sel}$.

The predictive performance of a rule set is measured with the micro-averaged F1-score. In the single-target setting, we measure the rule set's micro-averaged F1-score on the given target. In the multi-directional setting, we separately measure the micro-averaged F1-score on each target attribute and report the average.

To compare run time, we measure both the rule generation time and the rule selection time, the sum of which we call the total run time.

The model size of $\mathcal{R}_{sel}$ is indicated by three different metrics: (1) the number of literals in $\mathcal{R}_{sel}$ as the sum of its rule lengths, (2) its average rule length and (3) the number of rules in $\mathcal{R}_{sel}$.

Although the interpretability of a rule set is related to its model size, we also use three interpretability metrics as proposed for IDS [11]. First, we use $f_5(\mathcal{R})$ to measure the fraction of values occurring in the test data that are predicted by at least one rule (Section 6.2). Second, we consider the fraction of test set examples not covered by any rule. Third, we use the fraction of bodily overlap, which indicates how much the bodies of a rule set $\mathcal{R}$ overlap with respect to a test dataset of M instances, independent of the targets predicted by the rules:

$$\textit{fraction-bodily-overlap}(\mathcal{R}) = \frac{2}{|\mathcal{R}| \cdot (|\mathcal{R}| - 1)} \sum_{\substack{r_k, r_l \in \mathcal{R} \\ k<l}} \frac{|\textit{overlap}(r_k, r_l)|}{M}$$

7.2 Single-target tree rules vs. association rules.

Methodology To investigate **(Q1)** , we generate for both rule types a candidate single-target rule set of the same size. For both rule sets, we then use IDS to

select a single-target model. We call IDS using association rules *AR-IDS*, and using tree rules *T-IDS*. After rule selection, we compare the resulting models.

The two candidate rule sets are generated in two steps to ensure they have the same size. First, we generate the association rules using FP-growth [7] for a given support and confidence (instead of Apriori; see Sec. 6.1). Second, we learn a Scikit-learn Random Forest by increasing its number of trees until it corresponds to a rule set with the same number of rules or more. If it generates more rules, we sample without replacement as many tree rules as there are association rules. For both approaches, we use a minimum support of 0.1 and a maximum rule length of 7. For the tree rules, this corresponds to setting a minimum fraction of examples per leaf node of 0.1 and a maximum tree depth of 7.

For each dataset, we use two different candidate rule set sizes. We obtain these rule set sizes by using two different minimum confidence levels for the association rules: 0.75 and 0.95. Using a higher confidence results in a smaller candidate rule set (Fig. 1). Limiting the number of candidate rules is important because the computational cost of the rule selection step increases with the candidate rule set size. Note that other metrics than the confidence can be used to limit the number of association rules [17].

Results

Model size For a given candidate set size, the rule models selected by AR-IDS and T-IDS do not significantly differ in their number of rules or their number of literals (Fig. 1). Which of the two approaches selects more rules or contains more literals differs over the datasets. However, the average rule length is significantly shorter for tree rules than for association rules. For high confidence levels, this is to be expected, as association rules with a higher confidence are typically longer. But the tree rules are also shorter for confidence 0.75.

Run time The total run time seems independent of the rule type (as shown in Fig. 1). The rule generation time is negligible compared to the rule selection time, i.e. the time inducing an IDS model dominates over the rule generation time. Surprisingly, the time to generate association rules is not significantly different from the time to create tree rules. The rule selection time seems independent of the rule type, but increases with the candidate set size: selecting rules is much faster for confidence level 0.95 than for confidence level 0.75.

Predictive performance While we expected tree rules to lead to more accurate models than association rules, AR-IDS does not differ in micro-averaged F1-score from T-IDS in a statistically significant way. Our experiments also suggest there is no difference between both candidate set sizes.

Interpretability First, we see that rules selected by T-IDS have a significantly higher overlap than the rules selected by AR-IDS. Thus, tree rules require more tie-breaking. Second, while almost all examples in a test set are covered by the

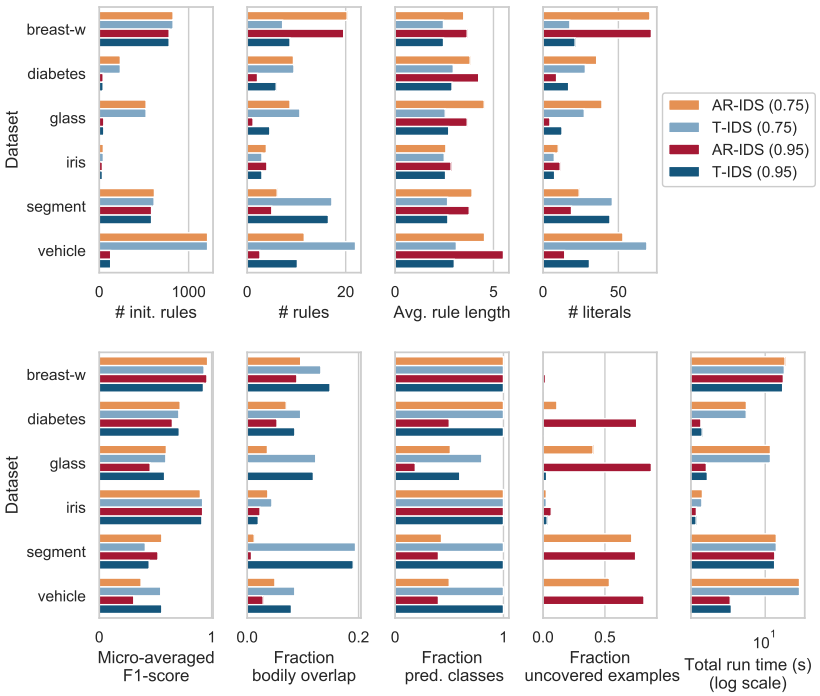

Fig. 1. Metrics quantifying the rule sets $\mathcal{R}_{sel}$ selected by IDS from single-target association rules and tree rules. For each dataset and rule type, two candidate rule set sizes were used by filtering the association rules on confidence 0.75 and 0.95.

T-IDS model, a large fraction is not covered by any rule in the AR-IDS model, thus requiring a default prediction without an explanation. Third, we see that T-IDS predicts more values in the target's domain than AR-IDS.

Discussion Answering **(Q1)**, comparing T-IDS and AR-IDS indicates that tree and association rules lead to rule sets that do not significantly differ in predictive performance, model size and run time. However, they differ in interpretability. First, the T-IDS models explain more predictions than AR-IDS models, as they cover more instances; AR-models fall back on unexplained default predictions more frequently. Second, the explained predictions are less clear for the T-IDS models than for the AR-IDS models, since the larger overlap indicates more rules have to be interpreted for a prediction. Third, the T-IDS models predict more values than AR-IDS. A possible explanation for the difference in coverage and overlap of the selected rule sets can be found in the similar difference in the candidate rule sets. The candidate tree rules also have a high overlap and coverage, as every point in the instance space is covered by as many rules as there are trees in the corresponding ensemble. In contrast, the candidate association rules do not have to cover the whole instance space, even though they can overlap.

Thus, our results suggest that for interpretability, tree rules are preferred for explaining predictions for as many instances as possible, or for having more class values explained by at least one rule. But if it is acceptable that a rule model cannot always make a prediction and might have to use a default value, association rules can give clearer predictions for the instances that are covered.

7.3 Multi-target vs. single-target rules

To investigate **(Q2)**, we compare single-target and multi-target rules in a multi-directional setting. We use tree rules, as their number is easy to control and our previous experiment indicates that tree rules and association rules lead to models similar in size and predictive performance. For the single-target rules, we use IDS to select one single-target IDS model per target and combine these models in a multi-directional ensemble called *eIDS*. For the multi-target rules, we use RR and MIDS as rule selectors.

Rule generation Both rules types are derived from Scikit-learn Random Forests using a minimum support of 0.1 and maximum rule length of 7.

For each attribute, a single-target candidate set is generated by (1) learning a Random Forest that predicts it, and (2) converting that Forest to rules. Each Random Forest contains 50 trees and has a maximum tree depth of 7.

One multi-target candidate set is constructed per dataset as follows. First, all attributes are randomly partitioned in groups of 2. For each group, a Random Forest of 10 trees is learned predicting those 2 attributes simultaneously. The attribute partitioning and Random Forest construction is repeated 5 times. As a result, each attribute is predicted by 5 Random Forests of 10 trees, or 50 trees in total. Then, one multi-target candidate set is generated for all target attributes by converting the trees of all Forest to rules. To ensure the rules have at most 7 literals, we use a maximum tree depth of 5 (as each tree predicts 2 targets).

Note that although each attribute is initially predicted by 50 trees in both the single-target and multi-target case, the number of rules predicting an attribute differs between the single-target and multi-target candidate rule sets. This results from each multi-target tree predicting 2 attributes. Thus, when combining the single-target rule sets, there are more candidate single-target rules than multi-target rules (*# init. rules* in Fig. 2).

Rule selection From each single-target candidate set, we use IDS to select a model. These single-target models are combined in one multi-directional ensemble model per dataset, called *eIDS*.

For each multi-target rule set, we use two rule selectors: RR and MIDS. For both RR and MIDS, one model is learned per dataset. We use RR with the same tie-breaking function and default predictions as used for (M)IDS, i.e. weighted voting and the majority class label. We set $\epsilon = 0.1$ and $\delta = 0.01$. (Sec. 5)

Results

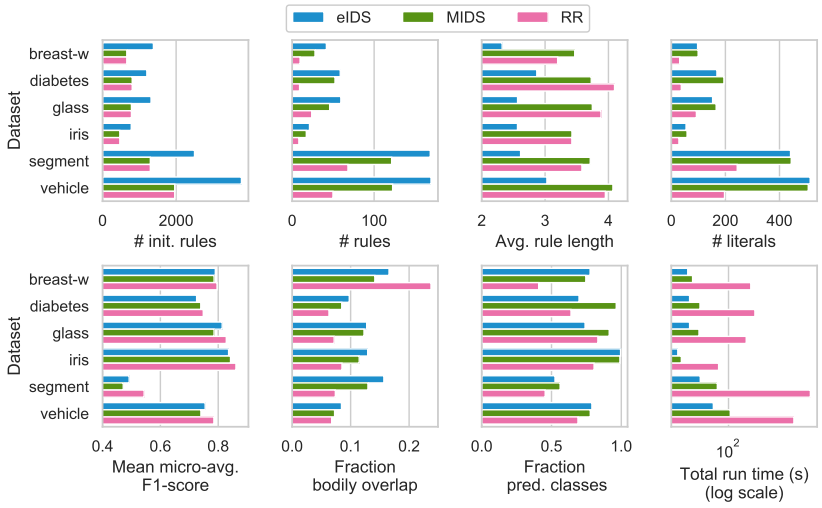

Fig. 2. Metrics quantifying the rule sets found using eIDS, MIDS and RR.

Run time For all three approaches, the rule generation time is negligible compared to the rule selection time. When comparing rule selection time, RR is orders of magnitude slower than eIDS and MIDS. Learning one MIDS model takes more time than learning the eIDS model, which can be explained by IDS selecting from smaller candidate sets and having a simpler objective function.

Model size RR results in a smaller model than both eIDS and MIDS (Fig. 2). The RR models contain significantly fewer literals than eIDS and MIDS, while the number of literals in the eIDS and MIDS models are similar.

When comparing the number of rules, the multi-target selection approaches result in the smallest rule sets. RR selects the smallest number of rules, while MIDS also selects significantly fewer rules than eIDS.

However, the multi-target selection approaches select significantly longer rules than eIDS. The average rule lengths of MIDS and RR are comparable, which can be expected, as they are built from the same candidate rule sets. The average rule lengths are longer for RR and MIDS than for eIDS, since the former use multi-target rules, whereas eIDS uses single-target rules.

Predictive performance While RR outperforms both eIDS and MIDS in micro-averaged F1-score, the micro-averaged F1-scores of eIDS and MIDS do not differ in a statistically significant manner.

Interpretability RR has a lower overlap than both MIDS and eIDS, while MIDS has a lower overlap than eIDS.

RR predicts fewer values occurring in the training data than eIDS and MIDS, between which there is no statistically significant difference.

As all three approaches cover almost all test instances with at least one rule, the fraction of uncovered instances is excluded from Fig. 2.

Discussion Answering **(Q2)**, our results for MIDS and eIDS indicate that in a multi-directional setting, learning a single model using multi-target rules instead of naively learning multiple single-target models can lead to fewer rules and less overlap between rules, but a similar predictive performance. A possible explanation is that the selected multi-target rules explicate correlations between different targets, which cannot occur in an ensemble of single-target rule models.

Our results also indicate it is better to use RR than MIDS or eIDS in a multi-target prediction setting. Unsurprisingly, RR outperforms (M)IDS in micro-averaged F1-score, as this is the only focus of RR, while the composite (M)IDS objective function also focuses on model size and interpretability. However, RR also outperforms (M)IDS on model size and interpretability. Not only does RR select rule sets with fewer rules and literals than the (M)IDS rule sets, RR also has the lowest rule overlap of the three approaches. The only benefits of using eIDS or MIDS over RR is that they are faster and their resulting rule sets provide explanations for a larger variety of values. This highlights it is often better to use a simple rule selection objective. Although it might be possible to find parameters λ_i for (M)IDS resulting in a similar model size and predictive performance as the RR models, this would require a potentially expensive hyperparameter optimization.

8 Conclusion

In this paper, we proposed how to train a multi-directional rule set based on multi-target tree rules, as a user might not know in advance which target will need to be predicted or which support and confidence thresholds to use with association rule mining. We proposed two new methods able to select multi-target rules: the greedy RR, focused on providing a high predictive performance on all targets, and MIDS, a generalization of IDS. Our experiments indicate that tree and association rules lead to models of similar size and predictive performance, although with different interpretability characteristics. Tree rules lead to models with a higher coverage, but association rules lead to clearer decision boundaries. We also showed that, compared to naively merging a collection of single-target rule models, using a multi-directional model built using multi-target rules results in fewer rules with lower overlap but with a similar predictive performance. Lastly, the usefulness of simple objective functions was demonstrated, as our RR models were not only more accurate than IDS and MIDS, they were also smaller with a lower overlap.

Future work While we compared single-target association and tree rules in the context of IDS, a similar comparison using other rule selection methods would be useful. Similarly, comparing RR and MIDS with other single-target rule selectors can help position these methods more clearly. Also, it would be interesting to generalize other rule selectors than IDS to handle multi-target rules.

Acknowledgments

This research received funding from the KU Leuven Research Fund (C14/17/070, "SIRV") and the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

1. Borgelt, C.: Frequent item set mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **2**(6), 437–456 (2012)
2. Bringmann, B., Nijssen, S., Zimmermann, A.: Pattern-Based Classification: A Unifying Perspective (2011)
3. Buchbinder, N., Feldman, M., Naor, J.S., Schwartz, R.: A tight linear time (1/2)-approximation for unconstrained submodular maximization. SIAM J. Comput. **44**(5), 1384–1402 (2015)
4. Feige, U., Mirrokni, V.S., Vondrák, J.: Maximizing Non-monotone Submodular Functions. SIAM J. Comput. **40**(4), 1133–1153 (2011)
5. Fürnkranz, J., Gamberger, D., Lavrač, N.: Foundations of Rule Learning. Springer (2014)
6. Fürnkranz, J., Knobbe, A.: Guest editorial: Global modeling using local patterns. Data Mining and Knowledge Discovery **21**(1), 1–8 (2010)
7. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. SIGMOD Rec. **29**(2), 1–12 (2000)
8. Ignatiev, A., Pereira, F., Narodytska, N., Marques-Silva, J.: A SAT-Based Approach to Learn Explainable Decision Sets. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) Automated Reasoning. pp. 627–645. Springer, Cham (2018)
9. Kliegr, T.: Quantitative CBA: Small and Comprehensible Association Rule Classification Models pp. 1–24 (2017)
10. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning. The MIT Press (2009)
11. Lakkaraju, H., Bach, S.H., Leskovec, J.: Interpretable Decision Sets: A Joint Framework for Description and Prediction. In: 22nd International Conference on Knowledge Discovery and Data Mining. pp. 1675–1684. KDD '16, ACM (2016)
12. Liu, B., Hsu, W., Ma, Y.: Integrating Classification and Association Rule Mining. In: Agrawal, R., Stolorz, P.E., Piatetsky-Shapiro, G. (eds.) 4th International Conference on Knowledge Discovery and Data Mining. pp. 80–86. AAAI Press (1998)
13. Quinlan, J.R.: Generating Production Rules from Decision Trees. In: McDermott, J.P. (ed.) 10th International Joint Conference on Artificial Intelligence. pp. 304–307. Morgan Kaufmann (1987)
14. Van Wolputte, E., Korneva, E., Blockeel, H.: MERCS: multi-directional ensembles of regression and classification trees. In: 32nd AAAI Conference on Artificial Intelligence. pp. 4276–4283 (2018)
15. Waegeman, W., Dembczyński, K., Hüllermeier, E.: Multi-target prediction: a unifying view on problems and methods. Data Mining and Knowledge Discovery **33**(2), 293–324 (2019)
16. Ženko, B., Džeroski, S.: Learning classification rules for multiple target attributes. In: 12th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2008). LNCS, vol. 5012, pp. 454–465. Springer (2008)
17. Zimmermann, A., De Raedt, L.: CorClass: Correlated association rule mining for classification. LNCS **3245**, 60–72 (2004)