Learning Semantically Coherent Rules

Alexander Gabriel[1], Heiko Paulheim[2], and Frederik Janssen[3]

[1] agabriel@mayanna.org
Technische Universität Darmstadt, Germany
[2] heiko@informatik.uni-mannheim.de
Research Group Data and Web Science
University of Mannheim, Germany
[3] janssen@ke.tu-darmstadt.de
Knowledge Engineering Group
Technische Universität Darmstadt, Germany

Abstract. The capability of building a model that can be understood and interpreted by humans is one of the main selling points of *symbolic* machine learning algorithms, such as rule or decision tree learners. However, those algorithms are most often optimized w.r.t. classification accuracy, but not the understandability of the resulting model. In this paper, we focus on a particular aspect of understandability, i.e., *semantic coherence*. We introduce a variant of a separate-and-conquer rule learning algorithm using a WordNet-based heuristic to learn rules that are semantically coherent. In an evaluation on different datasets, we show that the approach learns rules that are significantly more semantically coherent, without losing accuracy.

Keywords: Rule Learning, Semantic Coherence, Interpretability, Rule Learning Heuristics

1 Introduction

Symbolic machine learning approaches, such as rule or decision tree induction, have the advantage of creating a model that can be understood and interpreted by human domain experts – unlike statistical models such as Support Vector Machines. In particular, rule learning is one of the oldest and most intensively researched fields of machine learning [14].

Despite this advantage, the actual understandability of a learned model has received only little attention so far. Most learning algorithms are optimized w.r.t. the classification accuracy, but not understandability. Most often the latter is measured rather naively by, e.g., the average number of rules and/or conditions without paying any attention to the relation among them.

The understandability of a rule model comprises different dimensions. One of those dimensions is *semantic coherence*, i.e., the semantic proximity of the different conditions in a rule (or across the entire ruleset). Prior experiments have shown that this coherence has a major impact on the reception of a rule

In: P. Cellier, T. Charnois, A. Hotho, S. Matwin, M.-F. Moens, Y. Toussaint (Eds.): Proceedings of DMNLP, Workshop at ECML/PKDD, Nancy, France, 2014.

model. This notion is similar to the notion of semantic coherence of texts, which is a key factor to understanding those texts [20].

In a previous user study, we showed different rules describing the quality of living in cities to users. The experiments showed that semantically coherent rules – such as *Cities with medium temperatures and low precipitation* – are favored over incoherent rules, such as *Cities with medium temperatures where many music albums have been recorded* [27].

In this paper, we discuss how separate-and-conquer rule learning algorithms [12] can be extended to support the learning of more coherent rules. We introduce a new heuristic function that combines a standard heuristic (such as *Accuracy* or *m-Estimate*) with a semantic one, and allows for adjusting the weight of each component. With that weight, we are able to control the trade-off between classification accuracy and semantic coherence.

The rest of this paper is structured as follows. We begin by briefly introducing separate-and-conquer rule learning. Next, our approach to learning semantically coherent rules is detailed. In the following evaluation, we introduce the datasets and show the results. Here, also some exemplary rules are given, indeed indicating semantic coherence between the conditions of the rules. After that, related work is captured. Then, the paper is concluded and future work is shown.

2 Separate-and-Conquer Rule Learning

Separate-and-conquer rule learning is still amongst the most popular strategies to induce a set of rules that can be used to classify unseen examples, i.e., correctly map them on their respective classes. How exactly this strategy is implemented varies among the different algorithms but most of them fit into the framework of separate-and-conquer. This led to the development of the so-called SeCo suite [18], a versatile framework that allows for most existing algorithms to be configured properly. Based on the flexibility and the convenient way to implement new functions or extensions, we chose this framework for our experiments.

In essence, a separate-and-conquer rule learner proceeds in two major steps: First, a single rule, that fulfills certain quality criteria, is learned from the data (this is called the conquer step of the algorithm). Then, all (positive) examples that are covered by this rule are removed from the dataset (the separate step) and the algorithm proceeds by learning the next rule until all examples are covered.

Certainly, this strategy is only usable for binary data as a notion of positive and negative example is mandatory but then, if desired, it can guarantee that every positive example is covered (*completeness*) and no negative one is covered (*consistency*). There are different strategies to convert multi-class datasets to binary ones. However, in this paper we used an ordered binarization as implemented in the SeCo framework. Therefore, the classes of the dataset are ordered by their class-frequency and the smallest class is defined to be the positive one whereas the other ones are treated as negative examples. After the necessary number of rules to cover the smallest class is learned, all examples from it are removed and the next smallest one is defined to be positive while again the rest of the examples are negative. The algorithm proceeds in this manner until all classes expect the largest one are covered. The resulting ruleset is a so-called decision list where for each example that is to be classified the rules are tested from top to bottom and the first one that covers the example is used for prediction. If, however, no rule covers the example, a default rule at the end of the list assigns it to the largest class in the dataset.

A single rule is learned in a top-down fashion meaning that it is initialized as an empty rule and conditions are greedily added one by one until no more negative examples are covered. Then, the best rule encountered during this process is heuristically determined and returned as best rule. Note that this has not to be the last rule covering no negative example, i.e., *consistency* due to reasons of overfitting is not assured. A heuristic, in one way or another, maximizes the covered positive examples while trying to cover as few negative ones as possible. The literature shows a wide variety of different heuristics [13]. For the experiments conducted in this paper we had to make a selection and chose three well known heuristics namely *Accuracy*, *Laplace Estimate*, and the *m-Estimate*, as defined later. We are aware of the restrictions that come with our selection but we are confident that our findings regarding the semantic coherence are not subject to a certain type of heuristic but rather are universally valid.

To keep it simple, we used the default algorithm implemented in the SeCo framework. Namely, the configuration uses a top-down hill-climbing search (a beam size of one) that refines a rule as long as negative examples are covered. The learning of rules stops when the best rule covers more negative than positive examples and the conditions of a rule test for equality (nominal conditions) or use $<$ and $\geq$ for numerical conditions. No special pruning or post-processing of rules is employed. For the *m-Estimate*, the parameter was set to 22.466 as suggested in [17].

3 Enabling Semantic Coherence

The key idea of this paper is to enrich the heuristic used for finding the best condition with a semantic component that additionally to the goal of maximizing positive examples while minimizing negatives, will incorporate that the selected condition will also be as semantically coherent as possible. In essence, we have two components now:

- The classic heuristic (selects conditions based on statistical properties of the data) and
- the semantic heuristic (selects conditions based on their semantic coherence with previous conditions).

Hence, the new heuristic *WH* offers the possibility to trade-off between statistical validity (the classic heuristic *CH*) and the semantic part (a semantic heuristic *SH*). This is enabled by a parameter α that weights the two objectives:

$$WH(Rule) = \alpha \cdot SH(Rule) + (1 - \alpha) \cdot CH(Rule), \quad \alpha \in [0, 1] \qquad (1)$$

A higher value α gives more weight to semantic coherence, while a value of $\alpha = 0$ is equivalent to classic rule learning using only the standard heuristic. We expect that higher values of α lead to a decrease in predictive accuracy because the rule learning algorithm focuses less on the quality of the rule and more on choosing conditions that are semantically coherent (which are likely not to have a strong correlation with the rule's accuracy). At the same time, higher values of α should lead to more coherent rules.

When learning rules, the first condition is selected by using the classic heuristic *CH* only (since a rule with only one condition is always coherent in itself). Then, while growing the rule, the *WH* heuristic is used, which leads to conditions being added that result in both a coherent and an accurate rule according to the trade-off specified by α.

3.1 WordNet Similarity

There are different possibilities to measure the semantic relatedness between two conditions. In this paper, we use an external source of linguistic information, i.e., *WordNet* [8]. WordNet organizes words in so-called *synsets*, i.e., sets of synonym words. Those synsets are linked by homonym and hyperonym relations, among others. Using those relations, the semantic distance between words in different synsets can be computed.

In the first step, we map each feature that can be used in a rule to one or more synsets in WordNet[4]. To do so, we search WordNet for the feature name. In the following, we will consider the case of measuring the semantic coherence of two features named *smartphone vendor* and *desktop*.

The search for synsets returns a list of synsets, ordered by relevance. The search result for *smartphone vendor* is empty $\{\}$, the search result for *desktop* is $\{$*desktop#n#1*, *desktop#n#2*$\}$ where *desktop#n#1* describes a tabletop and *desktop#n#2* describes a desktop computer.[5]

If the list is not empty, we add it to the attribute label's list of synset lists. If otherwise the list is empty, we check whether the attribute label is a compound of multiple tokens and restart the search for each of the individual tokens. We then add all non-empty synset lists that are returned to the list of synset lists of the attribute label. The result for *smartphone vendor* is then $\{\{$*smartphone#n#1*$\}, \{$*vendor#n#1*$\}\}$ while the result for desktop is $\{\{$*desktop#n#1*, *desktop#n#2*$\}\}$.

In the second step, we calculate the distance between two synsets using the LIN [21] metric. We chose this metric as it performs well in comparison with other metrics [3], and it outputs a score normalized to $[0, 1]$.

[4] Note that at the moment, we only use the names of the features to measure semantic coherence, but not the nominal or numeric feature values that are used to build a condition.

[5] The 'n' indicates that the synsets are describing nouns.

The LIN metric is based on the Information Content (*IC*) metric [29], a measure for the particularity of a concept. The *IC* of a concept *c* is calculated as the negative of the log likelihood, simpler put: the negative of the logarithm of the probability to encounter concept *c*:

$$IC(c) = -\log p(c) \tag{2}$$

Higher values denote less abstract, more general concepts, while lower values denote more abstract, less general concepts. The body of text used for the calculation of the $p(c)$ values in this work is the SemCor [23] corpus, a collection of 100 passages from the Brown corpus which were semantically tagged "based on the WordNet word sense definition" and thus provide the exact frequency distribution of each synset, which covers roughly 25% of the synsets in WordNet [19].

The LIN metric is calculated by dividing the Information Content (*IC*) of the least common synset of the two synsets by the sum of their Information Content, and multiplying the result with two:[6]

$$lin(syn_1, syn_2) = 2 \cdot \frac{IC(lcs)}{IC(syn_1) + IC(syn_2)} \tag{3}$$

Information Content. For each pair of synsets associated with two attributes, we calculate the LIN metric. In our example, the corresponding values are

$$\begin{aligned} lin(smartphone\#n\#1, desktop\#n\#1) &= 0.0, \\ lin(smartphone\#n\#1, desktop\#n\#2) &= 0.5, \\ lin(vendor\#n\#1, desktop\#n\#1) &= 0.0, \text{ and} \\ lin(vendor\#n\#1, desktop\#n\#2) &= 0.0. \end{aligned}$$

In the third step, we choose the maximum value for each pair of synset lists (*syn*) so that we end up with the maximum similarity value per pair of tokens. The overall semantic similarity of two attributes (*att*) is then computed as the mean of those similarities across the tokens *t*:

$$SH(att_1, att_2) = \underset{\substack{\forall t_1 \in att_1 \\ \forall t_2 \in att_2}}{avg} \; \underset{\substack{\forall syn_1 \in t_1 \\ \forall syn_2 \in t_2}}{max} \; lin(syn_1, syn_2) \tag{4}$$

This assigns each word pair the similarity value of the synset combination that is most similar among all the synset combinations that arise from the two lists of possible synsets for the two words. Thus, in our example, the *SH* value assigned to *smartphone vendor* and *desktop* would be 0.25.

To compute the semantic coherence of a rule given the pairwise *SH* scores for the attributes used in the rule, we use the mean of those pairwise scores to assign a final score to the rule.[7]

[6] This metric limits the similarity calculation to synsets of the same POS and works only with nouns and verbs. Our implementation returns a similarity value of 0 in all other cases.

[7] All experiments were carried out with minimum and maximum as well, but using the mean turned out to give the best results.

Table 1. Datasets used in the experiments

Dataset	#Attributes	Found in WordNet
hepatitis	19	68%
primary-tumor	17	71%
bridges2	11	73%
zoo	17	94%
flag	27	100%
auto-mpg	7	100%
balloons	4	100%
glass	9	100%

4 Evaluation

We have conducted experiments with different classic heuristics on a number of datasets from the UCI machine learning repository[8] shown in Table 1. The table depicts the overall number of attributes and the percentage of attributes for which at least one matching synset was found in WordNet. For classic heuristics *CH*, we chose *Accuracy*, *m-Estimate*, and *Laplace Estimate*, which are defined as follows:

$$\mathit{Accuracy} := p - n \equiv \frac{p + (N - n)}{P + N} \tag{5}$$

$$\mathit{Laplace\ Estimate} := \frac{p + 1}{p + n + 2} \tag{6}$$

$$m\text{-}\mathit{Estimate} := \frac{p + m \cdot \frac{P}{P+N}}{p + n + m} \tag{7}$$

where p, n denote the positive/negative examples covered by the rule and P, N stand for the total positive/negative examples. Please see [17] for more details on these heuristics.

In addition, we used the semantic heuristic *SH* based on WordNet as defined above. For each experiment, we report the accuracy (single run of a ten fold cross validation) and the average semantic coherence of all the rules in the ruleset (measured by *SH*), as well as the average rule length and the overall number of conditions and rules in the ruleset.

As datasets, we had to pick some that have attribute labels that carry semantics, i.e., the attributes have to have speaking names instead of, e.g., names from `att1` to `att20` (which unfortunately is the case for the majority of datasets in the UCI repository). We searched for datasets where we could map at least two thirds of the attributes to at least one synset WordNet. This led to the eight datasets used for the experiments in this paper which are listed in Table 1.

Table 2. Macro average accuracy of the learned rulesets on the eight datasets. Statistically significant deviations ($p > 0.05$) from $\alpha = 0$ are marked in bold.

Classic	α										
Heuristic	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy	0.649	0.667	0.668	0.668	0.669	0.669	0.668	0.668	0.668	0.668	**0.465**
m-Estimate	0.670	0.673	0.672	0.671	0.671	0.670	0.670	0.673	0.673	0.674	**0.474**
Laplace	0.673	0.680	0.679	0.682	0.681	0.680	0.681	0.679	0.679	0.681	**0.476**

Table 3. Average semantic coherence of the learned rulesets on the eight datasets. Statistically significant deviations ($p > 0.05$) from $\alpha = 0$ are marked in bold.

Classic	α										
Heuristic	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy	0.146	**0.212**	0.222	**0.241**	**0.235**	**0.235**	**0.237**	**0.239**	**0.233**	**0.233**	–
m-Estimate	0.165	0.195	0.199	0.204	0.207	0.209	0.211	0.217	0.222	0.232	–
Laplace	0.156	**0.206**	**0.223**	**0.228**	**0.231**	**0.228**	**0.227**	**0.227**	**0.227**	**0.226**	–

4.1 Results of the Experiments

Table 2 shows the macro average accuracy across the eight datasets for different values of α. It can be observed that, except for $\alpha = 1$, the accuracy does not change significantly. This is an encouraging result, as it shows that a weight of up to 0.9 can be assigned to the semantic heuristic without the learning model losing much accuracy. How much exactly the coherence can be enforced has to be examined by a more detailed inspection of the parameter values in between 0.9 and 1.0. Interestingly, the trade-off between coherence and accuracy seems to occur rather at the edge at high parameter values. Clearly, a study of these parameters would yield more insights, but, however, ensuring such high coherence without a noticeable effect on accuracy already is a remarkable effect and seems to be sufficient for our purposes. Only when assigning all weight to the semantic heuristic (and none to the classic heuristic), the accuracy drops significantly, which is the expected result. In most of these cases, no rules are learned at all, but only a default rule is created, assigning all examples to the majority class.

In Table 3, we report the macro average semantic coherence of the learned rulesets across the eight datasets. The results have to be seen in context with Table 2 as our primary goal was to increase semantic coherence while not losing too much accuracy. Clearly, the higher the values of α will be, the more semantic coherence will be achieved anyway. This is because the heuristic component uses the same measure for semantic coherence as is reported in the evaluation in Table 3. However, as confirmation, it can be observed that the semantic coherence is indeed increased in all cases, whereas, when using *m-Estimate* as a classic heuristic, the increase is not statistically significant. As stated above, no

[8] http://archive.ics.uci.edu/ml/

Table 4. Two rules learned for primary-tumor

$\alpha = 0.0$	peritoneum = yes, skin = yes, histologic-type = adeno → class = ovary
$\alpha = 0.8$	peritoneum = yes, **skin = yes**, **pleura = no**, **brain = no** → class = ovary

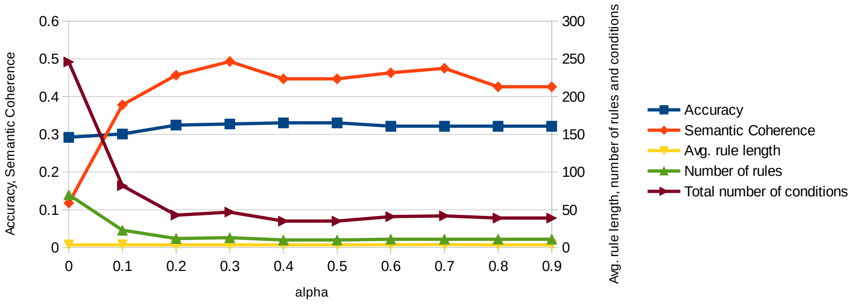

Fig. 1. Detailed results on the primary-tumor dataset, using *Accuracy* as a classic heuristic

rules are learned in many cases for $\alpha = 1$, so the semantic coherence cannot be computed there.

These results support our main claim, i.e., that it is possible to learn more coherent rules without losing classification accuracy. What is surprising is that even for $\alpha = 0.9$, the accuracy does not drop. This may be explained by the selection of the first condition in a rule, which is picked according only to the classic heuristic and thus leads to growing a rule that has at least a moderate accuracy. Furthermore, in many cases, there may be a larger number of possible variants for growing a rule the learning algorithm can choose from, each leading to a comparable value according to the classic heuristic, so adding weight to the semantic heuristic still can lead to a reasonable rule.

4.2 Analysis of the Models

The two rules learned for the primary-tumor dataset shown in Table 4 illustrate the difference between rules with and without semantic coherence. Both rules cover two positive and no negative example, i.e., according to any classic heuristic, they are equally good. However, the second one can be considered to be semantically more coherent, since three out of four attributes refer to body parts (skin, pleura, and brain), and are thus semantically related.

In order to further investigate the influence of the semantic heuristic on general properties of the learned ruleset, we also looked at the average rule length, the total number of rules, and the total number of conditions in a ruleset. The results are depicted in Tables 5 and 6.

In Table 5 we observe a mostly constant and sometimes increasing number of rules for all but the last three datasets. This exception to the overall trend is

Table 5. An overview of the number of rules and conditions in the learned rulesets for selected values of α for all datasets. Datasets where a drop occurred are shown at the end of the table.

Dataset		*Accuracy*		*m-Estimate*		*Laplace Estimate*	
	α	# rules	# conditions	# rules	# conditions	# rules	# conditions
	0.0	47	120	14	50	48	114
auto-mpg	0.5	48	127	14	46	47	110
	0.9	48	127	14	46	48	110
	0.0	2	4	2	4	4	12
balloons	0.5	2	4	2	4	4	12
	0.9	2	4	2	4	4	12
	0.0	27	59	10	25	30	65
bridges2	0.5	27	61	10	25	29	65
	0.9	27	61	10	25	29	65
	0.0	24	78	21	51	52	106
flag	0.5	38	90	24	63	54	113
	0.9	38	90	24	63	54	113
	0.0	12	14	11	15	19	19
zoo	0.5	16	18	12	16	19	19
	0.9	16	18	13	19	19	19
	0.0	40	90	16	52	50	102
glass	0.5	35	82	17	55	36	74
	0.9	35	82	17	55	36	74
	0.0	10	22	6	22	13	26
hepatitis	0.5	3	6	6	18	5	8
	0.9	3	6	6	18	5	8
	0.0	70	246	26	119	81	324
primary-tumor	0.5	10	35	12	56	75	331
	0.9	11	39	11	46	16	68

analyzed more closely in case of the primary-tumor dataset. The values for this dataset are depicted in Fig. 1.

When looking at the rulesets learned on the primary-tumor dataset, it can be observed that many very special rules for small classes, covering only a few examples, are missing when increasing the value for α. A possible explanation is that as long as there are many examples for a class, there are enough degrees of freedom for the rule learner to respect semantic coherence. If, on the other hand, the number of examples drops (e.g., for small classes), it becomes harder to learn meaningful semantic rules, which leads the rule learner to ignore those small example sets. Since only a small number of examples is concerned by this, the accuracy remains stable – or it even rises slightly, as ignoring those small sets may eventually reduce the risk of overfitting.

Note that a similar trend could be observed for the other two datasets (hepatitis and glass, depicted at the lower part of Table 5). While the changes are not so intense for the *m-Estimate*, certainly those for the other two heuristics are significant. Interestingly, most often the rules in the beginning of the decision list

Table 6. Average rule length of the learned rulesets on the eight datasets. Statistically significant deviations ($p > 0.05$) from $\alpha = 0$ are marked in bold.

Classic	α										
Heuristic	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy	2.270	2.295	2.296	2.296	2.281	2.281	2.310	2.321	2.288	2.288	**0.250**
m-Estimate	2.329	2.295	2.304	2.334	2.348	2.342	2.356	2.337	2.365	2.322	**0.250**
Laplace	2.920	2.851	2.862	2.811	2.811	2.833	2.828	2.821	2.796	2.788	**0.375**

are similar and at a certain point, no rules are learned any more. Thus, similar to the effect noticeable at the dataset primary-tumor, the following low coverage rules are not induced any more.

However, when looking at the average rule length (cf. Table 6), the only significant change occurs when all weight is given to the semantic component. The reason is that most often no rule is learned at all in this case.

4.3 Semantic Coherent Rules in Relation to Characteristic Rules

When we inspected the rule sets and the behavior of our separate-and-conquer learner in more detail, we found that semantically coherent rules interestingly have a connection to so-called characteristic rules [22, 4]. Where a discriminant rule tries to use as few conditions as possible with the goal of separating the example(s) of a certain class versus all the other ones, a characteristic rule has as much as possible conditions that actually describe the example(s) at hand. For instance, if the example to be described would be an elephant, a discriminant rule would concentrate on the single attribute(s) an elephant has and no other animal shows such as, e.g., its trunk, its gray color, or its huge ears. Instead, a characteristic rule would list all attributes that indicate an elephant such as four legs, a tail, thick skin etc. In essence, a discriminant rule has only conditions that discriminate elephants from all other animals whereas a characteristic rule rather describes the elephant without the need to be discriminant, i.e., to use only features no other animal has.

Not surprisingly, a semantically coherent rule tends to show the same properties. Often the induced rules consist of conditions that are not necessarily important to discriminate the examples, but rather are semantically coherent with the conditions located at earlier positions in these rules. This becomes obvious when we take a look at the above example of the two rules where the rule without semantic influence has a condition less albeit both of them have the same coverage.

However, the number of rules is strongly dependent on the attribute's semantics. For most of the datasets where actually less rules are induced with our approach, semantic coherence is hard to measure. The glass database contains of descriptions of chemicals, in the hepatitis dataset biochemical components are used as features and in primary-tumor we have simply considerably more classes. A detailed examination of this phenomenon remains subject to future work.

5 Related Work

Most of the work concerned with the trade-off between interpretability and accuracy stems from the fuzzy rules community. Here, this trade-off is well-known and there are a number of papers that addressed this problem [30]. There are several ways to deal with it, either by using (evolutionary) multiobjective optimization [16], context adaptation, hierarchical fuzzy modeling as well as fuzzy partitioning, membership functions, rules, rule bases or similar. However, most often comprehensibility of fuzzy rules is measured by means such as the transparency of the fuzzy partitions, the number of fuzzy rules and conditions or the complexity of reasoning, i.e., defuzzification and inference mechanisms. As we use classification rules in this paper, most of these techniques are not applicable.

There are also some papers about comprehensibility in general. For example, [33] deals with the means of dimensionality reduction and with presenting statistical models in a way that the user can grasp them better, e.g., with the help of graphical representations or similar. The interpretability of different model classes is discussed in [10]. The advantages and disadvantages of decision trees, classification rules, decision tables, nearest neighbor, and Bayesian networks are shown. Arguments are given why using model size on its own for measuring comprehensibility is not the best choice and directives are demonstrated how user-given constraints such as monotonicity constraints can be incorporated into the classification model. For a general discussion of comprehensibility this is very interesting, however, as single conditions of a rule are not compared against each other, the scope is somewhat different than in our work.

A lot of papers try to induce a ruleset that has high accuracy as well as good comprehensibility by employing genetic, evolutionary, or ant colony optimization algorithms. Given the right measure for relating single conditions of a rule or even whole rules in a complete ruleset, this seems to be a promising direction. Unfortunately, most of the fitness functions do not take this into account. For example, in [25] an extension of a ant colony algorithm was derived to induce unordered rulesets. They introduced a new measure for comprehensibility of rules, namely the *prediction-explanation size*. In essence this measure is oriented more strongly on the actual prediction hence the average number of conditions that have to be checked for predicting the class value. Therefore, not the total number of conditions or rules is measured as usual measures often do but for an unordered ruleset exactly those that are actually used for classifying the example at hand. For ordered rulesets also rules are counted that are before the classifying rule in the decision list as they have to be also checked at prediction time. Other algorithms are capable of multi-target learning [24] and define interestingness as those rules that cover example of infrequent classes in the dataset. Also, some papers deal with interpretability rather as a side effect [2], while here no optimization of this objective is done during learning time. In contrast, [7] uses a simple combination of accuracy maximization and size minimization in the fitness function of the genetic algorithm.

Some research is focused on specific problems where consequently rather unique properties are taken into account [31]. In this bioinformatic domain, only the presence of an attribute (value) is of interest whereas the absence is of no concern. The contribution are two new versions of CN2 [6] and Ant-Miner [26] which are able to incorporate this constraint.

Another thread is concerned with the measures themselves. For example, [9] surveyed objective measures (data-driven) for interestingness and defined a new objective, namely attribute surprisingness *AttSurp*, i.e., by arguing that a user is mostly interested in a rule that has high prediction performance but many single attributes with a low information gain, the authors define AttSurp as one divided by the information gain of all attributes in the rule. In [11] it is argued that small disjuncts (i.e., rules that cover only a very small number of positive examples) are indeed surprising while most often not unfolding good generalization or predictive quality. Here, also AttSurp is used which is different to most other interestingness measures in the sense that not the whole rule body is taken into account but single attributes which one can also see as a property of our algorithm. Interestingly, surprisingness also is related to Simpson's Paradox.

6 Conclusions and Future Work

In this paper, we have examined an approach to increase the understandability of a rule model by learning rules that are in themselves semantically coherent. To do so, we have introduced a method for combining classic heuristics, tailored at learning *correct* rule models, with semantic heuristics, tailored at learning *coherent* rules. While we have only looked at the coherence of *single* rules, adding means to control the coherence across a set of rules would be an interesting extension for future work.

An experiment with eight datasets from the UCI repository has shown that it is possible to learn rules that are significantly more coherent, while not being significantly less accurate. In fact, the accuracy of the learned model has stayed constant in all cases, even if adjusting the influence of the semantic heuristic to 90% of the overall heuristic. These results show that, even at a very preliminary stage, the proposed approach actually works.

Furthermore, we have observed that in some cases, adding the semantic heuristic may lead to more compact rule sets, which are still as accurate as the original ones. Although we have a possible explanation, i.e., that it is difficult for semantically enhanced heuristics to learn rules for small sets of examples, we do not have statistically significant results here. An evaluation with synthetic datasets may lead to more insights into the characteristics of datasets for which this property holds, and help us to confirm or reject that hypothesis.

Although we have evidence from previous research that semantically coherent rules are perceived to be better understandable, e.g. in [27], we would like to strengthen that argument by additional user studies. These may also help revealing other characteristics a ruleset should have beyond *coherence*, e.g., minimum or maximum length. For example, the experiments in [27] have indicated that less accurate rules (e.g., *Countries with a high HDI are less corrupt*) are preferred over more accurate ones (e.g., *Countries with a HDI higher than 6.243 are less corrupt*).

In this paper, we have only looked into one method of measuring semantic coherence, i.e., a similarity metric based on WordNet. There are more possible WordNet-based metrics, e.g., the LESK [1] and the HSO [15] metrics, which both work with adjectives and adverbs in addition to nouns and verbs and they support arbitrary pairing of the POS classes. Furthermore, there is a number of alternatives beyond WordNet, e.g., the use of Wikipedia [32] or a web search engine [5]. Furthermore, in the realm of Linked Open Data, there are various means to determine the relatedness of two concepts [28].

The approach so far only uses the classical heuristic to select the first rule, which sometimes lead to rules that are not too coherent w.r.t. that attribute, e.g., if there are no other attributes that match the first one well semantically. Here, it may help to introduce a semantic part in the selection of the first condition as well, e.g., the average semantic distance of all other attributes to the one selected. However, the impact of that variation on accuracy has to be carefully investigated.

Another possible point for improvement is the selection of the final rule from one refinement process. So far, we use the same combined heuristic for the refinement and the selection, but it might make sense to use a different weight here, or even entirely remove the semantic heuristic from that step, since the coherence has already been assured by the selection of the conditions.

In summary, we have introduced an approach that is able to explicitly trade off semantic coherence and accuracy in rule learning, and we have shown that it is possible to learn more coherent rules without losing accuracy. However, it remains an open question whether or not our results are generalizable to other types of rule learning algorithms that do not rely on a separate-and-conquer strategy. We will inspect the impact on other rule learners in the near future.

References

1. Banerjee, S., Pedersen, T.: An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In: Computational linguistics and intelligent text processing, pp. 136–145. Springer Berlin Heidelberg (2002)
2. Bojarczuk, C.C., Lopes, H.S., Freitas, A.A.: Discovering comprehensible classification rules by using genetic programming: a case study in a medical domain. In: Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M., Smith, R.E. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference. vol. 2, pp. 953–958. Morgan Kaufmann, Orlando, Florida, USA (1999)
3. Budanitsky, A., Hirst, G.: Evaluating WordNet-based Measures of Lexical Semantic Relatedness. Computational Linguistics 32(1), 13–47 (2006)
4. Cai, Y., Cercone, N., Han, J.: Attribute-oriented induction in relational databases. In: Knowledge Discovery in Databases, pp. 213–228. AAAI/MIT Press (1991)
5. Cilibrasi, R., Vitányi, P.M.B.: The google similarity distance. CoRR abs/cs/0412098 (2004)
6. Clark, P., Niblett, T.: The CN2 Induction Algorithm. Machine Learning 3(4), 261–283 (1989)

7. Falco, I.D., Cioppa, A.D., Tarantino, E.: Discovering interesting classification rules with genetic programming. Applied Soft Computing 1(4), 257 – 269 (2002)
8. Fellbaum, C.: WordNet. Wiley Online Library (1999)
9. Freitas, A.: On rule interestingness measures. Knowledge-Based Systems 12(56), 309 – 315 (1999)
10. Freitas, A.A.: Comprehensible classification models: A position paper. SIGKDD Explor. Newsl. 15(1), 1–10 (Mar 2014)
11. Freitas, A.A.: On objective measures of rule surprisingness. In: Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery. pp. 1–9. PKDD '98, Springer-Verlag, London, UK, UK (1998)
12. Fürnkranz, J.: Separate-and-Conquer Rule Learning. Artificial Intelligence Review 13(1), 3–54 (1999)
13. Fürnkranz, J., Flach, P.A.: ROC 'n' Rule Learning - Towards a Better Understanding of Covering Algorithms. Machine Learning 58(1), 39–77 (January 2005)
14. Fürnkranz, J., Gamberger, D., Lavrač, N.: Foundations of Rule Learning. Springer Berlin Heidelberg (2012)
15. Hirst, G., St-Onge, D.: Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, pp. 305–332. MIT Press (1995)
16. Ishibuchi, H., Nojima, Y.: Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. International Journal of Approximate Reasoning 44(1), 4–31 (Jan 2007)
17. Janssen, F., Fürnkranz, J.: On the quest for optimal rule learning heuristics. Machine Learning 78(3), 343–379 (Mar 2010)
18. Janssen, F., Zopf, M.: The SeCo-framework for rule learning. In: Proceedings of the German Workshop on Lernen, Wissen, Adaptivität - LWA2012 (2012)
19. Jiang, J.J., Conrath, D.W.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: Proceedings of International Conference on Research in Computational Linguistics (ROCLING X). pp. 19–33. No. Rocling X (1997)
20. Kintsch, W., Van Dijk, T.A.: Toward a model of text comprehension and production. Psychological review 85(5), 363 (1978)
21. Lin, D.: An Information-Theoretic Definition of Similarity. In: ICML. pp. 296–304 (1989)
22. Michalski, R.S.: A theory and methodology of inductive learning. Artificial Intelligence 20(2), 111–162 (1983)
23. Miller, G.a., Leacock, C., Tengi, R., Bunker, R.T.: A Semantic Concordance. In: Proceedings of the workshop on Human Language Technology. pp. 303–308. Association for Computational Linguistics, Morristown, NJ, USA (1993)
24. Noda, E., Freitas, A., Lopes, H.: Discovering interesting prediction rules with a genetic algorithm. In: Proceedings of the 1999 Congress on Evolutionary Computation. pp. 1322–1329. IEEE (1999)
25. Otero, F.E., Freitas, A.A.: Improving the interpretability of classification rules discovered by an ant colony algorithm. In: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation. pp. 73–80. GECCO '13, ACM, New York, NY, USA (2013)
26. Parpinelli, R.S., Lopes, H.S., Freitas, A.A.: Data mining with an ant colony optimization algorithm. IEEE Transactions on Evolutionary Computation 6(4), 321–332 (August 2002)
27. Paulheim, H.: Generating possible interpretations for statistics from linked open data. In: 9th Extended Semantic Web Conference (ESWC) (2012)

28. Paulheim, H.: Dbpedianyd – a silver standard benchmark dataset for semantic relatedness in dbpedia. In: Workshop on NLP & DBpedia (2013)
29. Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence. vol. 1 (1995)
30. Shukla, P.K., Tripathi, S.P.: A survey on interpretability-accuracy (i-a) trade-off in evolutionary fuzzy systems. In: Watada, J., Chung, P.C., Lin, J.M., Shieh, C.S., Pan, J.S. (eds.) 5th International Conference on Genetic and Evolutionary Computing. pp. 97–101. IEEE (2011)
31. Smaldon, J., Freitas, A.A.: Improving the interpretability of classification rules in sparse bioinformatics datasets. In: Proceedings of AI-2006, the Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence. pp. 377–381. Research and Development in Intelligent Systems XXIII, Springer London (2007)
32. Strube, M., Ponzetto, S.P.: WikiRelate! Computing semantic relatedness using Wikipedia. In: In Proceedings of the 21st National Conference on Artificial Intelligence. pp. 1419–1424. No. February, AAAI Press (2006)
33. Vellido, A., Martn-Guerrero, J.D., Lisboa, P.J.G.: Making machine learning models interpretable. In: 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (2012)