Evaluating Algorithms for Concept Description

Cecilia Sönströd, Ulf Johansson and Tuve Löfström

Abstract—**When performing concept description, models need to be evaluated both on accuracy and comprehensibility. A comprehensible concept description model should present the most important relationships in the data in an accurate and understandable way. Two natural representations for this are decision trees and decision lists. In this study, the two decision list algorithms RIPPER and Chipper, and the decision tree algorithm C4.5, are evaluated for concept description, using publicly available datasets. The experiments show that C4.5 performs very well regarding accuracy and brevity, i.e. the ability to classify instances with few tests, but also produces large models that are hard to survey and contain many extremely specific rules, thus not being good concept descriptions. The decision list algorithms perform reasonably well on accuracy, and are mostly able to produce small models with relatively good predictive performance. Regarding brevity, Chipper is better than RIPPER, using on average fewer conditions to classify an instance. RIPPER, on the other hand, excels in relevance, i.e. the ability to capture a large number of instances with every rule.**

I. Introduction

In the data mining task concept description [1], the aim is to gain insights. The focus is not to produce models with high predictive accuracy, but to adequately describe the most important relationships in the data. Recommended techniques for this task are rule induction and conceptual clustering. In many concept description situations, what is actually needed is a highly understandable classification model, indicating that rule induction is most suitable. However, most rule induction algorithms focus on optimizing predictive performance, i.e. accuracy, often at the expense of comprehensibility, and few, if any, include direct ways for the user to control the tradeoff between accuracy and comprehensibility. Of course, the term comprehensibility is far from unproblematic, since many factors influence whether a model is understandable or not. Often, model size is used to estimate comprehensibility, with the implicit assumption that small models are easier to interpret.

In [2], we proposed that concept description models should be evaluated using accuracy and three properties that capture different aspects of comprehensibility. Hence, the four criteria for evaluating concept description models become:

- Accuracy: The model should have high accuracy on unseen data to guarantee that relationships found hold in general
- Brevity: The model should classify as many instances as possible with few conditions
- Interpretability: The model should express conditions in a clear and readable way
- Relevance: Only those relationships that are general and interesting should be included in the model

The most common representation language for transparent models is decision trees, and many successful decision tree algorithms exist. Looking at the above criteria, it is clear that standard decision tree algorithms such as C4.5 [3] and CART [4] are capable of fulfilling the accuracy criterion. Intuitively, the decision tree representation also seems suitable for obtaining good brevity, since a balanced decision tree will make many different classifications with relatively few conditions used for each path to a leaf node. However, regarding the two other criteria, decision trees have some serious drawbacks, since the representation leads to models that are hard to survey, and typically also contain many branches that classify only a few instances. Looking at a decision tree, especially in its textual representation, a decision-maker will have a hard time finding and interpreting the most important relationships; indeed, he would probably trace the sequence of tests and write out a separate rule, consisting of a conjunction of tests, for branches of interest.

The above example is an argument for the alternative representation decision lists, or ordered rule sets. A decision list is, in essence, a maximally unbalanced decision tree, where each rule, containing one or more tests, directly classifies a number of instances. Those instances not covered by one rule are tested on the next rule, and this proceeds until a default rule classifies all remaining instances. This representation is especially suitable for concept description, since it admits a rule construction method prioritizing high coverage for the top rules, thereby obtaining good brevity and relevance by capturing the most general relationships in the dataset with very few conditions; or, put in another way, describing the most important relationships with very simple rules. Furthermore, since the top rules in a decision list are very easy to identify, a decision list algorithm with high coverage in its top rules will also have good interpretability. Finally, the default rule used in decision lists also provides a mechanism for increasing both interpretability and relevance, since models can avoid formulating rules that only classify a few instances.

All authors are with the CSL@BS at the School of Business and Informatics, Unversity of Borås, Sweden. Tuve Löfström is also with the School of Humanities and Informatics, University of Skövde, Sweden.
e-mail: {cecilia.sonstrod, ulf.johansson, tuve.lofstrom}@hb.se

Ultimately, a decision list algorithm suitable for concept description should consistently obtain accuracy and brevity comparable to standard decision tree algorithms, with its representation offering benefits for interpretability and relevance. It must be noted that this is a tough challenge, since industrial-strength decision trees are very good at optimizing accuracy and has a representation that lends itself very well to good brevity

II. Background

Most decision list algorithms are based on the sequential covering algorithm, which in each step constructs a rule covering a number of instances and then removes those instances from the dataset before the next rule is found. This procedure is repeated until some stop criterion is met, when a default rule, classifying all remaining instances, is formulated. Examples of decision list algorithms based on this idea are AQ [5], CN2 [6], IREP[7] and RIPPER [8].

RIPPER (*Repeated Incremental Pruning to Produce Error Reduction*) is widely used and considered to be the most successful decision list algorithm to date [9], and is reported to have good performance regarding both accuracy [10], running time and ability to handle noise and unbalanced data sets [11]. RIPPER works by constructing rules only for the minority class(es) and using the default rule for the majority class. For multiclass problems, rules are constructed for classes in order of class distribution. In [9], most of RIPPER's power is attributed to its optimization (post-pruning) procedure and the authors argue that post-pruning is probably the best way to obtain good predictive performance for decision list algorithms.

However, some of the properties that give RIPPER good performance regarding accuracy and speed are detrimental to its concept description ability. First, and foremost, for binary problems with relatively even class distributions, rules are only formulated for the class that happens to have the lowest number of instances, and hence no explicit description (other than as the negation of the rules for the minority class) is offered for instances belonging to the other class. This will also often result in RIPPER obtaining relatively poor brevity even for quite short rule sets. Furthermore, RIPPER is built to optimize accuracy and will sometimes do so at the expense of comprehensibility, without any clear possibility for the user to control this tradeoff via parameter settings.

We have previously, see [12], introduced the decision list algorithm Chipper, aimed at performing concept description. Chipper is a greedy algorithm based on sequential covering, and the basic idea is to, in each step, find the rule that classifies the maximum number of instances using only one split on one attribute. The algorithm uses a parameter, called *ignore*, to control the tradeoff between accuracy and coverage. This parameter specifies the acceptable misclassification rate for each rule, expressed either as an absolute number of instances or as a proportion of remaining instances in the data set. The *ignore* parameter can thus be used to control the granularity of the rules produced, with high *ignore* values being used for finding rules with high coverage (possibly with lower rule accuracy) and low values for more specific rules. The other main parameter in Chipper is called *stop* and is used to determine the proportion of instances that should be described by rules before the default rule is formulated. Thus, a *stop* value of 80% means that the default rule will be formed when at least 80% of the instances are covered by the rule set.

In [12], Chipper was evaluated on 9 binary datasets from the UCI machine learning repository [13], which were chosen to enable interpretation of rule sets found, and compared to two standard algorithms for generating transparent models; RIPPER and C4.5. Results were very encouraging, with Chipper obtaining accuracy comparable to other techniques, but having superior comprehensibility.

However, it was noted that Chipper sometimes performed very badly regarding accuracy and also was very sensitive to the value of the *ignore* parameter. This is, of course, not entirely a bad thing, since it means that the parameter works as intended, letting the user choose the granularity of the rule set. In some cases, though, it is of course desirable to have the option of finding "the best possible" model without having to manually search for optimal parameter settings.

The main purpose of this study is to conduct a more thorough evaluation of both a slightly improved version of Chipper, but also RIPPER and C4.5 regarding suitability for concept description. It is of course interesting to investigate how existing algorithms perform regarding brevity and relevance, especially for decision trees which are not normally evaluated on these criteria.

III. Method

The main change in Chipper from [12] is that the algorithm is able to handle multiclass problems and is now implemented in Java and incorporated in the WEKA [9] data mining framework. Implementation in WEKA has facilitated solving the problem with excessive sensitivity to the value of the *ignore* parameter, by using the built-in meta procedure for cross-validation parameter selection (CVParameterSelection). CVParameterSelection uses internal cross-validation to find the best values for one or more parameters within a specified range. Obviously, most standard techniques, such as RIPPER and C4.5, use an internal cross-validation procedure as a part of their algorithm to optimize models on accuracy. Using this procedure to set parameter values in Chipper gives the desired flexibility regarding parameter use, since the user can either specify a given range or set specific values.

Finally, using techniques all implemented in the same framework means an improved ability to carry out comparisons by using controlled experiments with fixed folds for all techniques.

A. Datasets

26 publicly available dataset from the UCI repository [13] were used for the experiments. As seen in Table 1 below, where the dataset characteristics are presented, both binary and multiclass problems are used, and the datasets have different properties regarding number of instances, as well as the number of continuous (*Cont.*) and categorical (*Cat.*) attributes.

TABLE 1. DATA SETS USED

Data set	Instances	Classes	Cont.	Cat.
Breast cancer	286	2	0	9
Cmc	1473	3	2	7
Colic – Horse	368	2	7	15
Credit – American	690	2	6	9
Credit – German	1000	2	7	13
Cylinder	540	2	18	21
Diabetes – Pima	768	2	8	0
E-coli	336	8	7	0
Glass	214	6	9	0
Haberman	306	2	2	1
Heart – Cleveland	303	2	6	7
Heart – Statlog	270	2	10	3
Hepatitis	155	2	6	13
Ionosphere	351	2	34	0
Iris	150	3	4	0
Labor	57	2	8	8
Liver disorders – BUPA	345	2	6	0
Lymphography	148	4	3	15
Sick	3772	2	22	7
Sonar	208	2	60	0
Tae	151	3	3	2
Vehicle	846	4	18	0
Votes	435	2	0	16
Wine	178	3	13	0
WBC - Wisconsin	699	2	9	0
Zoo	101	7	16	1

B. Experiments

In the experiments, Chipper is compared to two techniques producing transparent models, but using different representations. The chosen decision tree algorithm is again C4.5 implemented as J48 in WEKA and the decision list technique is RIPPER, implemented as JRip in WEKA. The motivation for these choices is that the two algorithms are standard techniques for their respective representations.

In the first experiment, the aim is to investigate whether Chipper is able to obtain acceptable predictive performance, measured as accuracy and *area under the ROC-curve* (AUC), over a large number of datasets. The motivation for including both accuracy and AUC is that they measure quite different things; while accuracy is based only on the final classification, AUC measures the ability of the model to rank instances according to how likely they are to belong to a certain class; see e.g. [14]. AUC can be interpreted as the probability of ranking a true positive instance ahead of a false positive; see [15].

In this experiment, Chipper is used with parameter settings favoring accuracy, using CVParameterSelection for both for *stop* and *ignore*. *Ignore* had values between 0.5% and 5%, with 10 different values and *stop* was between 70% and 95%, with 6 different values. In this experiment, 10 x 10-fold cross-validation is used, with identical folding for all three techniques, for measuring accuracy and AUC. J48 and JRip were used with their default settings in WEKA, which means that J48 trees are pruned.

In the second experiment, the tradeoff between accuracy and comprehensibility is investigated by using Chipper with settings favoring comprehensible models, meaning lower *stop* values and higher *ignore* values. The choice was to use *stops* between 60% and 80%, and *ignores* between 4% and 8%. For this kind of rule set, where the aim is to present the most important relationships in the data, the evaluation should primarily be on comprehensibility, with overall model accuracy serving only to guarantee that the relationships found will hold in general.

C. Evaluation of comprehensibility

For measuring the brevity aspect of comprehensibility, we have previously, in [2], suggested the *classification complexity* (CC) measure. The classification complexity for a model is the average number of tests (i.e. simple conditions using only one attribute) needed to classify an instance. A low value thus means good brevity.

The interpretability and relevance aspects, however, have not as yet been formulated as numeric measures. The most problematic of these is arguably interpretability; in Chipper, interpretability is ensured by the very simple representation language, allowing only one test in each rule. For relevance, a measure would have to differentiate between models containing few and many rules, but also be more refined than just model size. A model should be deemed to have high relevance when every part of the model classifies a large number of instances. To make comparison between different representations possible, we introduce the term *classification point*, and define it as a place where instances are assigned class labels in a classification model. For a decision list, the classifications points then consist of all rules, and for a decision tree, every leaf is a classification point. We propose that relevance can be measured by calculating the average number of instances that reach each classification point in a model. Thus, a high value will represent good relevance. This in essence, means calculating the load factor for each rule or leaf. This measure will simply be called *relevance* and is calculated using (1) below:

$$relevance = \frac{\#instances\ in\ dataset}{\#classification\ points} \quad (1)$$

Comprehensibility for the models produced in both experiments is consequently evaluated using classification complexity (CC), and the relevance measure introduced above.

IV. RESULTS

The results regarding accuracy and AUC from Experiment 1 are shown in Table 2 below. Bold numbers indicate the best result for a specific dataset.

TABLE 2. RESULTS ACCURACY AND AUC, 10X10CV

	J48		JRip		Chipper	
Data set	**Acc**	**AUC**	**Acc**	**AUC**	**Acc**	**AUC**
Breast-cancer	70.5%	0.59	**71.5%**	**0.60**	67.8%	0.54
Cmc	52.3%	**0.69**	**52.4%**	0.64	48.2%	0.66
Colic	**85.3%**	**0.85**	85.1%	0.83	77.6%	0.78
Credit - A	**85.2%**	0.87	**85.2%**	0.87	85.1%	**0.91**
Credit - G	70.6%	0.64	**72.2%**	0.63	70.1%	**0.65**
Cylinder	**73.2%**	**0.76**	67.1%	0.66	66.9%	0.71
Diabetes	74.5%	0.75	75.2%	0.72	**76.0%**	**0.79**
E-coli	**82.8%**	**0.96**	81.4%	0.95	71.1%	0.95
Glass	**67.6%**	0.79	66.8%	**0.80**	61.5%	0.73
Haberman	72.2%	0.58	72.4%	0.60	**76.3%**	**0.61**
Heart – C	78.2%	0.78	**80.0%**	**0.81**	71.2%	0.74
Heart – S	78.2%	0.79	**78.8%**	**0.80**	72.0%	0.73
Hepatitis	79.2%	0.67	78.1%	0.62	**80.8%**	**0.74**
Ionosphere	**89.7%**	**0.89**	89.2%	**0.89**	87.2%	0.85
Iris	**94.7%**	**0.99**	93.9%	**0.99**	93.1%	**0.99**
Labor	78.4%	0.72	**83.7%**	**0.82**	78.9%	0.73
Liver	65.8%	**0.65**	**66.6%**	**0.65**	63.2%	0.63
Lymph	75.6%	**0.77**	76.3%	0.40	**79.2%**	0.46
Sick	**98.9%**	**0.96**	98.3%	0.94	96.5%	0.94
Sonar	73.6%	0.75	73.4%	0.75	**75.5%**	**0.78**
Tae	**57.4%**	**0.75**	43.8%	0.60	42.1%	0.60
Vehicle	**72.3%**	0.76	68.3%	**0.77**	59.9%	0.73
Vote	**96.6%**	**0.98**	95.8%	0.96	94.7%	0.97
Wine	**93.2%**	**0.97**	93.1%	0.96	91.9%	**0.97**
WBC	95.0%	**0.96**	**95.6%**	**0.96**	93.8%	0.95
Zoo	**91.6%**	**1.00**	86.6%	0.93	87.9%	**1.00**
#Wins	13	14	9	10	5	9

As can be seen from the table, J48 performs best overall, both regarding accuracy and AUC. J48 obtains the highest accuracy on 13 out of 25 datasets, and JRip performs slightly better than Chipper. However, for some datasets, there are significant differences in accuracy, with Chipper quite often losing by a large margin. This is probably due to there being no global rule set optimization targeting accuracy in Chipper, whereas the two other techniques spend quite a lot of effort on post-pruning and optimizing their models on accuracy, using internal cross-validation.

When measuring predictive performance using AUC, J48 still stands out, winning 14 datasets, albeit 5 of them drawn with other techniques. In contrast to accuracy, there is very little difference between Chipper and JRip on the AUC measure.

The comprehensibility results from Experiment 1 are shown in Table 3 below. Classification complexity (*CC*) is given for the number of tests, *size* is measured simply as the total number of tests in the rule set, serving as an indicator of overall model complexity, and relevance (*Rel*) is calculated according to (1) above. All these measures are given for the model constructed over the entire dataset that WEKA outputs.

When calculating CC, everything using more than 30 conditions is lumped together as if it were in the default rule; the only dataset where this happens is Vehicle for both JRip and Chipper. Obviously, J48 has an advantage for the CC measure, because of its different representation.

TABLE 3. COMPREHENSIBILITY RESULTS FOR EXPERIMENT 1

Data set	**J48**			**JRip**			**Chipper**		
	Size	**CC**	**Rel**	**Size**	**CC**	**Rel**	**Size**	**CC**	**Rel**
Breast-c	21	5.1	13	**4**	**4.6**	**95**	8	7.8	32
Cmc	131	**8.9**	11	14	12.6	**295**	**13**	10.0	105
Colic	**5**	**1.7**	61	6	5.4	**92**	13	5.0	26
Credit - A	24	3.8	28	**7**	6.0	**173**	**7**	**2.4**	86
Credit - G	98	9.2	10	11	5.3	**333**	**8**	**5.1**	111
Cylinder	66	23	8	**6**	**6.0**	**180**	22	9.9	23
Diabetes	19	**3.7**	38	**9**	8.0	**192**	10	3.8	70
E-coli	21	**4.4**	15	19	16.7	**34**	**13**	4.8	24
Glass	29	**5.9**	7	**18**	14.9	**31**	25	13.1	8
Haberman	10	**2.3**	28	**3**	3.9	**153**	12	7.1	24
Heart – C	17	**3.6**	17	**6**	5.3	**76**	11	6.7	25
Heart – S	17	**3.0**	15	8	6.7	**54**	**7**	4.2	34
Hepatitis	10	**2.9**	14	**5**	5.3	**39**	6	3.2	22
Ionosphere	17	5.8	20	**2**	**2.4**	**117**	13	7.7	25
Iris	4	**2.1**	30	**3**	2.3	**38**	**3**	2.2	**38**
Labor	6	**2.7**	8	**4**	3.7	**19**	5	3.2	10
Liver	25	4.7	13	**4**	**4.2**	**115**	13	7.7	25
Lymph	13	**5.0**	11	**8**	7.5	**25**	13	5.3	11
Sick	22	3.2	164	10	9.5	943	**2**	**1.4**	**1257**
Sonar	17	4.4	12	9	7.6	**42**	**5**	**3.2**	35
Tae	33	6.4	4	**1**	**1.8**	**76**	20	9.0	7
Vehicle	97	**7.2**	9	**43**	23.1	**53**	49	18.3	17
Vote	5	**1.6**	73	6	5.2	**145**	**3**	**1.6**	109
Wine	**4**	**2.3**	36	**4**	3.9	**60**	5	2.5	30
WBC	13	2.7	50	9	7.4	**140**	**4**	**1.9**	**140**
Zoo	8	2.8	11	**6**	5.8	**20**	**6**	**2.7**	**14**
#Wins	2	15	0	17	5	25	10	7	4
Average	28.2	4.9		8.7	7.1		11.4	5.8	

As can be seen from the table, J48 is able to use its representation to obtain very good classification complexity, despite having by far the largest average model size. However, J48 performs very badly on relevance, not winning a single dataset. This, together with the very good classification complexity, indicates that the decision trees produced by J48 contain a lot of branches that classify very few instances each, which means that the overall comprehensibility of the model is limited.

Turning to a direct comparison between Chipper and RIPPER, the results are very clear. Chipper is superior to RIPPER on classification complexity, performing better on 19 out of 26 datasets and also obtaining a lower average number of tests. It is, of course, slightly iffy to calculate an average like this, but what this average says is that over all datasets, Chipper needs just under six tests to classify an instance. JRip performs best on relevance, winning all but one datasets, which of course is a consequence of the smaller model size. Indeed, RIPPER rule sets typically consist of between 3 and 4 rules and seldom contain rules classifying only a few instances.

Below are some sample rule sets for Chipper and JRip, where the differences between how the two techniques work become very apparent.

```
IF Color_intensity <= 3.4 THEN 1 [55/0]
IF Proline >= 885.0 THEN 0 [49/0]
IF Flavanoids <= 1.22 THEN 2 [42/0]
IF Magnesium <= 94.0 THEN 1 [11/0]
IF Proline >= 680.0 THEN 0 [10/0]
DEFAULT: 2 [11/5]
```

Figure 1: Chipper sample rule set for Wine

```
IF Flavanoids <= 1.39 AND
  Color_intensity >= 3.85 THEN 3 [47/0]
IF Proline >= 760 AND
  Color_intensity >= 3.52 THEN 1 [57/0]
DEFAULT: 2 [74/3]
```

Figure 2: JRip sample rule set for Wine

```
IF FATIGUE == 0.0 THEN LIVE [54/2]
IF PROTIME >= 51.0 THEN LIVE [34/1]
IF ALBUMIN >= 4.1 THEN LIVE [12/1]
IF BILIRUBIN >= 3.5 THEN DIE [8/1]
IF SEX == 0.0 THEN LIVE [7/0]
IF ALBUMIN <= 2.6 THEN DIE [5/0]
DEFAULT: LIVE [35/16]
```

Figure 3: Chipper sample rule set for Hepatitis

```
IF ALBUMIN <= 3.8 AND
  ALBUMIN <= 2.8 THEN DIE [13/2]
IF PROTIME <= 42 THEN DIE [15/7]
IF SPIDERS = yes AND
  BILIRUBIN >= 2 THEN DIE [11/4]
DEFAULT: LIVE [116/6]
```

Figure 4: JRip sample rule set for Hepatitis

In the Hepatitis rule set, Chipper's emphasis on brevity is very clear, with top rules covering many more instances than in the corresponding JRip rule set. For a concept description task, this is clearly a desirable property. For Wine, JRip builds just one rule for each class, obtaining very good relevance, but slightly worse classification complexity than Chipper, due to having the top rule for the minority class and using multiple conditions in both rules.

The results from Experiment 2, where Chipper is set to optimize comprehensibility, are shown in Table 4 below. For comparison between the two settings, Chipper accuracies from Experiment 1 are included in the table.

TABLE 4. CHIPPER ACCURACY AND SIZE RESULTS FOR EXPERIMENT 2

	Chipper Exp 1		Chipper Exp 2	
Data set	**Acc**	**Size**	**Acc**	**Size**
Breast-cancer	67.9%	8	**68.7%**	**5**
Cmc	48.2%	13	**51.1%**	**12**
Colic	**77.7%**	13	76.8%	**2**
Credit - A	**85.7%**	7	85.3%	**2**
Credit - G	70.1%	8	**71.0%**	**7**
Cylinder	**69.0%**	22	68.2%	**16**
Diabetes	**76.7%**	10	73.1%	**4**
E-coli	**71.7%**	13	67.3%	**9**
Glass	60.3%	25	**62.6%**	**17**
Haberman	74.3%	12	**74.9%**	**9**
Heart – C	71.1%	11	**73.5%**	**4**
Heart – S	**71.6%**	**7**	69.5%	**7**
Hepatitis	**82.6%**	**6**	82.2%	**6**
Ionosphere	87.4%	13	**87.9%**	**2**
Iris	**92.7%**	**3**	88.7%	**3**
Labor	**80.6%**	5	74.2%	**2**
Liver	**63.4%**	13	61.7%	**9**
Lymph	**80.1%**	13	74.1%	**9**
Sick	**97.8%**	**2**	94.5%	**2**
Sonar	**76.7%**	5	65.2%	**2**
Tae	42.5%	20	**48.6%**	**7**
Vehicle	**60.7%**	49	57.2%	**5**
Vote	95.1%	3	**95.8%**	**2**
Wine	**94.1%**	5	88.3%	**3**
WBC	**95.0%**	4	91.3%	**1**
Zoo	**90.9%**	6	88.7%	**4**
#Wins	17	4	9	26
Average		11.4		5.8

Here, there are several striking results. First of all, Chipper actually obtains higher accuracy on 9 datasets with settings prioritizing short and clear rule sets. This is obviously an indication that there maybe is a problem with over-fitting when using the settings optimizing accuracy. It is also notable, but not surprising, that a considerable loss of accuracy occurs for datasets where accuracy is very high, see e.g. the WBC, Wine and Iris datasets, since Chipper is now allowed to formulate much less accurate rules.

Regarding rule set size, rules are on average much smaller for these Chipper settings; indeed, rule set size is decreased for all but 4 datasets. A really good example of how a considerable reduction in rule set size only leads to a relatively small loss of accuracy is seen in the Vehicle dataset, where an almost incomprehensible model consisting of 49 rules is replaced by a much more comprehensible 5-rule model, with only a small loss in accuracy.

```
IF HOLLOWS RATIO <= 186
       THEN 2 [147/60]
IF SCALED VARIANCE_MINOR <= 310
       THEN 3 [133/54]
IF DISTANCE CIRCULARITY <= 76
       THEN 2 [125/40]
IF ELONGATEDNESS >= 42
       THEN 3 [111/29]
IF MAX.LENGTH RECTANGULARITY >= 169
       THEN 0 [77/26]
DEFAULT: 1 [253/132]
```

Figure 5: Sample Chipper rule set for Vehicle

The comprehensibility measures for Experiment 2 are given below in Table 5. Again, Chipper and JRip results from Experiment 1 are included for reference.

TABLE 5. BREVITY AND RELEVANCE RESULTS FOR EXPERIMENT 2

Data set	JRip		Chipper Experiment 1		Chipper Experiment 2	
	CC	Rel	CC	Rel	CC	Rel
Breast-cancer	4.6	**95**	7.8	32	**3.6**	48
Cmc	12.6	**295**	10.0	105	**8.6**	113
Colic	5.4	92	5.0	26	**1.7**	**123**
Credit - A	6.0	173	2.4	86	**1.8**	**230**
Credit - G	5.3	**333**	5.1	111	**3.7**	125
Cylinder	**6.0**	**180**	9.9	23	8.4	32
Diabetes	8.0	**192**	3.8	70	**2.7**	154
E-coli	16.7	**34**	4.8	24	**4.6**	**34**
Glass	14.9	**31**	13.1	8	**10.0**	12
Haberman	3.9	**153**	7.1	24	**5.6**	31
Heart – C	5.3	**76**	6.7	25	**2.8**	61
Heart – S	6.7	**54**	**4.2**	34	**4.2**	34
Hepatitis	5.3	**39**	**3.2**	22	**3.2**	22
Ionosphere	2.4	**117**	7.7	25	**1.9**	**117**
Iris	2.3	**38**	2.2	38	**2.0**	**38**
Labor	3.7	**19**	3.2	10	**2.0**	**19**
Liver	**4.2**	**115**	7.7	25	6.0	35
Lymph	7.5	**25**	**5.3**	11	6.1	15
Sick	9.5	943	1.4	**1257**	**1.3**	**1257**
Sonar	7.6	42	3.2	35	**2.0**	**70**
Tae	**1.8**	**76**	9.0	7	4.4	19
Vehicle	23.1	53	18.3	17	**3.7**	**141**
Vote	5.2	**145**	1.6	109	**1.5**	**145**
Wine	3.9	**60**	2.5	30	**2.1**	45
WBC	7.4	140	1.9	140	**1.3**	**350**
Zoo	5.8	**20**	2.7	14	**2.5**	**20**
#Wins	3	20	3	1	22	12
Average	7.1		5.8		3.8	

The overall picture in this experiment is that the reduction in size carries over to classification complexity, i.e. that these shorter rules significantly improve the ability to classify a majority of instances with few conditions. The same holds for relevance, where Chipper is now significantly better, but still not as good as RIPPER.

Below are some sample Chipper rules from Experiment 2, once again illustrating that rules have good brevity, relevance and interpretability.

```
IF plas <= 107.0 THEN 0 [289/36]
IF plas >= 156.0 THEN 1 [117/23]
IF mass <= 28.2 THEN 0 [102/18]
IF preg >= 8.0 THEN 1 [44/11]
DEFAULT: 0.0[216/87]
```

Figure 6: Sample Chipper rule set for Diabetes

```
IF physician-fee-freeze == 0
       THEN 0 [247/2]
IF synfuels-corporation-cutback == 0
       THEN 1 [139/3]
DEFAULT: 1 [49/19]
```

Figure 7: Sample Chipper rule set for Vote

V. CONCLUSIONS

The results show that there is a definite tradeoff between accuracy and comprehensibility when performing concept description, with no technique or representation being superior in all areas. C4.5 produces very high predictive performance, and also uses its tree representation to obtain very good classification complexity, which means that C4.5 trees are accurate with good brevity. However, these models are often quite big and contain many relationships that only hold for a few instances, and which are of very limited value in a concept description task. Indeed, these spurious components may even obscure the important relationships found in the data by making the model hard to survey and interpret.

RIPPER performs quite well on accuracy and also manages to use its representation with conjunctive rules to obtain very good relevance, with typical models containing few rules, each with high coverage. Unfortunately, RIPPER is inherently unable to really perform well on brevity, i.e. classification complexity, since rules are only formulated for the minority classes. Although this is a clear advantage for unbalanced problems where the minority class is the one of interest, it is in general not suitable for concept description.

Chipper, finally, manages to obtain competitive accuracy on some datasets, even though it does not contain any post-processing of rule to optimize accuracy on the whole model. Chipper clearly outperforms RIPPER on brevity, especially when using parameter settings favoring comprehensibility, which reduce both model size and classification complexity, without significant loss of accuracy. In fact, over all datasets, Chipper models in Experiment 2 use only 4 conditions on average to classify an instance.

Overall, the conclusion is that both decision trees and decision lists have some properties that are suitable for concept description, but that no algorithm performs well on all evaluated criteria. Clearly, some work remains to handle the tradeoff between predictive performance and different aspects of comprehensibility when performing concept description.

VI. DISCUSSION AND FUTURE WORK

Since Chipper, although designed for concept description, has proved capable of sometimes obtaining accuracy on par with techniques aimed at predictive performance, it would of course be interesting to see if Chipper could be extended with the aim of increasing predictive accuracy, but keeping the core idea of greedy search for rules with high coverage. It is then natural to extend the representation language to include intervals, conjunctions and disjunctions. An argument for this was seen in [16], where Chipper was evaluated on a medical dataset and where it became apparent that the representation language, allowing only a single condition per rule, was too simple to capture some of the

important relationships in the dataset. Another possibility, when aiming to increase predictive power, is to use different rule selection criteria, more capable of handling the tradeoff between coverage and accuracy. Obviously, when seeking to improve predictive accuracy, the issue of over-fitting must be handled, especially since the present study indicates that this is already a problem when Chipper is used with prediction settings.

There are of course many possible extensions that would improve concept description ability. It could perhaps be argued that the current *ignore* parameter, although having a very marked effect on rules produced, is not all that intuitive. Since it sort of depends on the number of instances in the dataset, it is quite hard to predict what kind of rule accuracy a specific *ignore* value will produce. For decision-makers, a parameter formulated directly in terms of rule accuracy would be easier to use efficiently, especially since domain constraints (such as cost of false positives and negatives) often can be translated into rule accuracy. Obviously, in this context, rule selection could be based on precision and/or recall instead of accuracy, with the ability to prioritize one class.

To a lesser extent, the non-intuitive reservation also applies to the *stop* parameter. An alternative way of formulating this would be that the user could just specify the desired number of rules before the default rule is applied.

When looking at the results and rule sets from the Sick dataset, which is extremely unbalanced with 3772 instances and the class distribution of 3541/231, it is obvious that RIPPER's approach of finding rules only for the minority class and then applying the default rule to classify the remaining instances works extremely well. With such dataset, the minority class will also almost always be the one that decision-makers are interested in obtaining a concept description of. Hence, a natural extension to Chipper is to let the user determine explicitly which class(es) should be described by rules and which class(es) should only be captured by the default rule.

REFERENCES

[1] The CRISP-DM Consortium, CRISP-DM 1.0, www.crisp-dm.org, 2000.

[2] C. Sönströd, U. Johansson and R. König, "Towards a Unified View on Concept Description", *The 2007 International Conference on Data Mining (DMIN07)*, Las Vegas, NV, 2007.

[3] J. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufman, 1993.

[4] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*, Wadsworth International Group, 1984.

[5] J. Hong, I. Mozetic, and R.S. Michalski, "AQ15 : Incremental learning of attribute-based descriptions from examples, the method and user's guide", Report ISG 85-5, UIUCDCS-F-86-949, Dept. of Computer Science, University of Illinois at Urbana-Champaign, 1986.

[6] P. Clark and T. Niblett, "The CN2 induction algorithm", *Machine Learning*, 3:261-283, 1989.

[7] J. Fürnkrantz and G. Widmer, "Incremental Reduced Error Pruning", *Proceedings of the 11th International Conference on Machine Learning (ICML'94),* p.70-77, San Mateo, CA, 1994.

[8] W. Cohen, "Fast Effective Rule Induction", *Proceedings of the 12th International Conference on Machine Learning (ICML'95)*, p. 115-123. Tahoe City, CA, 1995.

[9] J. Fürnkrantz and P. Flach, "An Analysis of Stopping and Filtering Criteria for Rule Learning", *Proceedings of the 15th European Conference on Machine Learning*, 123–133, 2004.

[10] I. H. Witten and E. Frank, *Data Mining – Machine Learning Tools and Techniques*, 2nd ed, Morgan Kaufman, 2005.

[11] P. Tan, M. Steinbach and V. Kumar, *Introduction to Data Mining*, Pearson Education, 2006.

[12] U. Johansson, C. Sönströd, T. Löfström and H. Boström, "Chipper – A Novel Algorithm for Concept Description", *10th Scandinavian Conference on Artificial Intelligence*, IOS Press, pp. 133-140, Stockholm, Sweden, 2008.

[13] C. L. Blake and C. J. Merz, UCI Repository of Machine Learning Databases, University of California, Department of Information and Computer Science, 1998.

[14] T. Fawcett, "Using rule sets to maximize roc performance", *15th International Conference on Machine Learning*, pp. 445-453, 2001

[15] A. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms", *Pattern Recognition*, 30:1145-1159, 1997.

[16] C. Sönströd, U. Johansson, U. Norinder, and H. Boström, "Comprehensible Models for Predicting Molecular Interaction with Heart-Regulating Genes" , *7th International Conference on Machine Learning and Applications (ICMLA '08)*, Orlando, FL, IEEE press, pp. 559 – 564, 2008.