APRIORI-SD: Adapting association rule learning to subgroup discovery

Conference Paper *in* Applied Artificial Intelligence · June 2008

DOI: 10.1007/978-3-540-45231-7_22

Branko KavŠek
University of Primorska
28 PUBLICATIONS **895** CITATIONS
SEE PROFILE

Nada Lavrac
Jožef Stefan Institute
415 PUBLICATIONS **11,009** CITATIONS
SEE PROFILE

APRIORI-SD: Adapting Association Rule Learning to Subgroup Discovery

Branko Kavšek, Nada Lavrač, and Viktor Jovanoski

Institute Jožef Stefan, Jamova 39, 1000 Ljubljana, Slovenia
{branko.kavsek,nada.lavrac,viktor.jovanoski@ijs.si}

Abstract. This paper presents a subgroup discovery algorithm APRIORI-SD, developed by adapting association rule learning to subgroup discovery. This was achieved by building a classification rule learner APRIORI-C, enhanced with a novel post-processing mechanism, a new quality measure for induced rules (weighted relative accuracy) and using probabilistic classification of instances. Results of APRIORI-SD are similar to the subgroup discovery algorithm CN2-SD while experimental comparisons with CN2, RIPPER and APRIORI-C demonstrate that the subgroup discovery algorithm APRIORI-SD produces substantially smaller rule sets, where individual rules have higher coverage and significance.

1 Introduction

Classical rule learning algorithms are designed to construct classification and prediction rules [12,3,4,7]. In addition to this area of machine learning, referred to as *supervised learning* or *predictive induction*, developments in *descriptive induction* have recently gained much attention, in particular *association rule learning* [1] (e.g., the APRIORI association rule learning algorithm), *subgroup discovery* (e.g., the MIDOS subgroup discovery algorithm [18,5]), and other approaches to non-classificatory induction.

As in the MIDOS approach, a subgroup discovery task can be defined as follows: given a population of individuals and a property of those individuals we are interested in, find population subgroups that are statistically 'most interesting', e.g., are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest [18,5].

Some of the questions on how to adapt classical classification rule learning approaches to subgroup discovery have already been addressed in [10] and a well-known rule learning algorithm CN2 was adapted to subgroup discovery. In this paper we take a rule learner APRIORI-C instead of CN2 and adapt it to subgroup discovery, following the guidelines from [10].

We have implemented the new subgroup discovery algorithm APRIORI-SD in C++ by modifying the APRIORI-C algorithm. The proposed approach performs subgroup discovery through the following modifications of the rule learning algorithm APRIORI-C: (a) using a weighting scheme in rule post-processing, (b) using weighted relative accuracy as a new measure of the quality of the rules in

M.R. Berthold et al. (Eds.): IDA 2003, LNCS 2810, pp. 230–241, 2003.

the post-processing step when the best rules are selected, (c) probabilistic classification based on the class distribution of covered examples by individual rules, and (d) area under the ROC curve rule set evaluation. The latter evaluation criterion was used in addition to the standard evaluation criteria: rule coverage, rule significance, rule set size and rule set accuracy.

This paper presents the APRIORI-SD subgroup discovery algorithm, together with its experimental evaluation in selected domains of the UCI Repository of Machine Learning Databases [13]. Results of APRIORI-SD are similar to the subgroup discovery algorithm CN2-SD while experimental comparisons with CN2, RIPPER and APRIORI-C demonstrate that the subgroup discovery algorithm APRIORI-SD produces substantially smaller rule sets, where individual rules have higher coverage and significance. These three factors are important for subgroup discovery: smaller size enables better understanding, higher coverage means larger subgroups, and higher significance means that rules describe subgroups whose class distribution is significantly different from the entire population by no loss in terms of the area under the ROC curve and accuracy.

This paper is organized as follows. In Section 2 the background for this work is explained: the APRIORI-C rule induction algorithm, including the post-processing step of selecting the best rules. Section 3 presents the modified APRIORI-C algorithm, called APRIORI-SD, adapting APRIORI-C for subgroup discovery together with the weighted relative accuracy measure, probabilistic classification and rule evaluation in the ROC space. Section 4 presents the experimental evaluation on selected UCI domains. Section 5 concludes by summarizing the results and presenting plans for further work. While in Section 4 we present the summary results of the experiments, the complete results on all the UCI data sets are presented in the Appendix.

2 Background: The APRIORI-C Algorithm

The idea of using association rules for classification is not new [11]. The main advantage of APRIORI-C over its predecessors is lower memory consumption, decreased time complexity and improved understandability of results. The reader can find a detailed description of the APRIORI-C algorithm in [7]. We describe here just the parts of the APRIORI-C that are essential for the reader to understand the derived APRIORI-SD algorithm.

The APRIORI-C algorithm is derived from the well known association rule learning algorithm APRIORI [1,2] which was extensively studied, adopted to other areas of machine learning and data mining, and successfully applied in many problem domains.

An association rule has the following form:

$$X \to Y, \tag{1}$$

where $X, Y \subset I$, X and Y are itemsets, and I is the set of all items.

The quality of each association rule is defined by its *confidence* and *support*. *Confidence* of a rule is an estimate of the conditional probability of Y given X:

$p(Y|X)$. *Support* of a rule is an estimate of the probability of itemset $X \cup Y$: $p(XY)$. Confidence and support are computed as follows:

$$Confidence = \frac{n(XY)}{n(X)} = \frac{p(XY)}{p(X)} = p(Y|X), Support = \frac{n(XY)}{N} = p(XY) \quad (2)$$

where $n(X)$ is the number of transactions that are supersets of itemset X and N is the number of all the transactions.

The association rule learning algorithm APRIORI is then adopted for classification purposes (APRIORI-C) by implementing the following steps: (1) Discretize continuous attributes. (2) Binarize all (discrete) attributes. (3) Run the APRIORI algorithm by taking in consideration only rules whose right-hand side consists of a single item, representing the value of the class attribute (while running APRIORI). (4) Post-process this set of rules, selecting the best among them and use this rules to classify unclassified examples.

These steps of the APRIORI-C algorithm, as well as the approaches to feature subset selection, are described in detail in [7]. Here we describe just the last step, the post-processing of rules and classification of unclassified examples, which are the ones we changed to obtain APRIORI-SD.

Post-processing by rule subset selection. The APRIORI-C algorithm induces rules according to the parameters *minimal confidence* and *minimal support* of a rule [7]. The setting of these two parameters is often such that the algorithm induces a large number of rules, which hinders the understandability and usability of induced rules. Moreover, there are the problems of rule redundancy, incapability of classifying examples and poor accuracy in domains with unbalanced class distribution. A way to avoid these problems is to select just some best rules among all the induced rules. APRIORI-C has three ways of selecting such best rules:

Use N best rules: The algorithm first selects the best rule (rule having the highest support), then eliminates all the covered examples, sorts the remaining rules according to support and repeats the procedure. This procedure is repeated until N rules are selected or there are no more rules to select or there are no more examples to cover. The algorithm then stops and returns the classifier in the form of an IF-THEN-ELSE rule list.

Use N best rules for each class: The algorithm behaves in a similar way as the 'use N best rules' case, selecting N best rules for each class (if so many rules exist for each class). This way the rules for the minority class will also find their way into the classifier.

Use a weighting scheme to select the best rules: The algorithm again behaves in a similar way as 'use N best rules'. The difference is that covered examples are not eliminated immediately, but instead their weight is decreased. They are then eliminated when the weight falls below a certain threshold (e.g., when an example has been covered more than K times). The details of the weighting scheme together with the threshold used are given in Section 3, describing APRIORI-SD.

3 APRIORI-SD

The main modifications of the APRIORI-C algorithm, making it appropriate for subgroup discovery, involve the implementation of a new weighting scheme in post-processing, a different rule quality function (the weighted relative accuracy), the probabilistic classification of unclassified examples and the area under the ROC curve rule set evaluation.

3.1 Post-processing Procedure

The post-processing procedure is performed as follows:

```
repeat
- sort rules from best to worst in terms of
  the weighted relative accuracy quality measure (see Section 3.3)
- decrease the weights of covered examples (see Section 3.2)
until
  all the examples have been covered or
  there are no more rules
```

3.2 The Weighting Scheme Used in Best Rule Selection

In the 'use a weighting scheme to select best rules' post-processing method of APRIORI-C described in Section 2, the examples covered by the 'currently' best rule are not eliminated but instead re-weighted. This approach is more suitable for the subgroup discovery process which is, in general, aimed at discovering interesting properties of subgroups of the entire population. The weighting scheme allows this.

The weighting scheme treats examples in such a way that covered positive examples are not deleted when the currently 'best' rule is selected in the post-processing step of the algorithm. Instead, each time a rule is selected, the algorithm stores with each example a count that shows how many times (with how many rules selected so far) the example has been covered so far. Initial weights of all positive examples e_j equal 1, $w(e_j, 0) = 1$, which denotes that the example has not been covered by any rule, meaning 'among the available rules select a rule which covers this example, as this example has not been covered by other rules', while lower weights mean 'do not try too hard on this example'.

Weights of positive examples covered by the selected rule decrease according to the formula $w(e_j, i) = \frac{1}{i+1}$. In the first iteration all target class examples contribute the same weight $w(e_j, 0) = 1$, while in the following iterations the contributions of examples are inverse proportional to their coverage by previously selected rules. In this way the examples already covered by one or more selected rules decrease their weights while rules covering many yet uncovered target class examples whose weights have not been decreased will have a greater chance to be covered in the following iterations.

3.3 The Weighted Relative Accuracy Measure

Weighted relative accuracy is used in subgroup discovery to evaluate the quality of induced rules. We use it instead of support when selecting the 'best' rules in the post-processing step.

We use the following notation. Let $n(X)$ stand for the number of instances covered by a rule $X \rightarrow Y$, $n(Y)$ stand for the number of examples of class Y, and $n(YX)$ stand for the number of correctly classified examples (true positives). We use $p(YX)$ etc. for the corresponding probabilities. Rule accuracy, or rule confidence in the terminology of association rule learning, is defined as $Acc(X \rightarrow Y) = p(Y|X) = \frac{p(YX)}{p(X)}$. Weighted relative accuracy [9,16] is defined as follows.

$$WRAcc(X \rightarrow Y) = p(X).(p(Y|X) - p(Y)). \tag{3}$$

Weighted relative accuracy consists of two components: generality $p(X)$, and relative accuracy $p(Y|X) - p(Y)$. The second term, relative accuracy, is the accuracy gain relative to the fixed rule *true* → Y. The latter rule predicts all instances to satisfy Y; a rule is only interesting if it improves upon this 'default' accuracy. Another way of viewing relative accuracy is that it measures the utility of connecting rule body X with a given rule head Y. However, it is easy to obtain high relative accuracy with highly specific rules, i.e., rules with low generality $p(X)$. To this end, generality is used as a 'weight', so that weighted relative accuracy trades off generality of the rule ($p(X)$, i.e., rule coverage) and relative accuracy ($p(Y|X) - p(Y)$). All the probabilities in Equation 3 are estimated with relative frequencies e.g., $p(X) = \frac{n(X)}{N}$, where N is the number of all instances.

Modified WRAcc function with Example Weights. WRAcc - the rule quality function used in APRIORI-SD was further modified to enable handling example weights, which provide the means to consider different parts of the instance space with each application of a selected rule (as described in Section 3.2).

The modified *WRAcc* measure is defined as follows:

$$WRAcc(X \rightarrow Y) = \frac{n'(X)}{N'}\left(\frac{n'(YX)}{n'(X)} - \frac{n'(Y)}{N'}\right). \tag{4}$$

where N' is the sum of the weights of all examples, $n'(X)$ is the sum of the weights of all covered examples, and $n'(YX)$ is the sum of the weights of all correctly covered examples.

3.4 Probabilistic Classification

In general the induced rules can be treated as ordered or unordered. Ordered rules are interpreted as an IF-THEN-ELSE decision list [15] in a straight-forward manner: when classifying a new example, the rules are sequentially tried and the first rule that covers the example is used for prediction. This interpretation of rules is also used by APRIORI-C, when classifying unclassified examples.

In the case of unordered rule sets, the distribution of covered training examples among classes is attached to each rule. Rules of the form:

$$X \rightarrow Y \; [ClassDistribution]$$

are induced, where numbers in the *ClassDistribution* list denote, for each individual class, the percentage of training examples of this class covered by the rule. When classifying a new example, all rules are tried and those covering the example are collected. If a clash occurs (several rules with different class predictions cover the example), a voting mechanism is used to obtain the final prediction: the class distributions attached to the rules are considered to determine the most probable class. If no rule fires, a default rule is invoked which predicts the majority class of uncovered training instances. This type of interpretation of induced rules for classification, also called probabilistic classification, is used by the APRIORI-SD algorithm.

3.5 Area under ROC Curve Evaluation

A point on the *ROC curve* (ROC: Receiver Operating Characteristic) [14] shows classifier performance in terms of false alarm or *false positive rate* $FPr = \frac{FP}{TN+FP} = \frac{FP}{Neg}$ (plotted on the X-axis; 'Neg' standing for the number of all negative examples) that needs to be minimized, and sensitivity or *true positive rate* $TPr = \frac{TP}{TP+FN} = \frac{FP}{Pos}$ (plotted on the Y-axis; 'Pos' standing for the number of all positive examples) that needs to be maximized. The confusion matrix shown in Table 1 defines the notions of TP (true positives), FP (false positives), TN (true negatives) and FN (false negatives).

Table 1. Confusion matrix.

	predicted positive	predicted negative
actual positive	TP	FN
actual negative	FP	TN

Applying the notation used to define *confidence* and *support* (see Equation 2) FPr and TPr can be expressed as: $FPr = \frac{n(X\overline{Y})}{Neg}$, $TPr = \frac{n(XY)}{Pos}$. In the ROC space, an appropriate tradeoff, determined by the expert, can be achieved by applying different algorithms, as well as by different parameter settings of a selected data mining algorithm or by taking into the account different misclassification costs. The ROC space is appropriate for measuring the success of subgroup discovery, since subgroups whose TPr/FPr tradeoff is close to the diagonal can be discarded as insignificant. The reason is that the rules with TPr/FPr on the diagonal have the same distribution of covered positives and negatives as the distribution in the entire data set.

The area under the ROC curve (*AUC*) can be used as a quality measure for comparing the success of different learners. AUC can be computed by employing combined probabilistic classifications of all subgroups (rules) [10], as indicated below. If we always choose the most likely predicted class, this corresponds to setting a fixed threshold 0.5 on the positive probability: if the positive probability is larger than this threshold we predict positive, else negative. A ROC curve can be constructed by varying this threshold from 1 (all predictions negative, corresponding to (0,0) in the ROC space) to 0 (all predictions positive, corresponding to (1,1) in the ROC space). This results in $n + 1$ points in the ROC space, where n is the total number of classified examples. Equivalently, we can order all the testing examples by decreasing predicted probability of being positive, and tracing the ROC curve by starting in (0,0), stepping up when the tested example is actually positive and stepping to the right when it is negative, until we reach (1,1); in the case of ties, we make the appropriate number of steps up and to the right at once, drawing a diagonal line segment. The area under this ROC curve indicates the combined quality of all subgroups (i.e., the quality of the entire rules set). This method can be used with a test set or in cross-validation. A detailed description of this method can be found in [6].

This method of computing the AUC was used in the experimental evaluation in Section 4.

4 Experimental Evaluation

For subgroup discovery, expert's evaluation of results is of ultimate interest. Nevertheless, before applying the proposed approach to a particular problem of interest, we wanted to verify our claims that the mechanisms implemented in the APRIORI-SD algorithm are indeed appropriate for subgroup discovery.

We experimentally evaluated our approach on 23 data sets from the UCI Repository of Machine Learning Databases [13]. In Table 2, the selected data sets are summarized in terms of the number of attributes, number of classes, number of examples, and the percentage of examples of the majority class. All continuous attributes were discretized with a discretization method described in [8] using the WEKA tool [17].

The comparison of APRIORI-SD with APRIORI-C, and RIPPER was performed in terms of coverage of rules, size of the rule sets, significance of rules, area under the ROC curve and predictive accuracy (despite the fact that optimizing predictive accuracy is not the intended goal of subgroup discovery algorithms). The area under the ROC curve evaluation was computed only on two-class problems (first 16 data sets in Table 2). The method we used for evaluation was 10-fold stratified cross validation. The parameters used to run the algorithms APRIORI-SD and APRIORI-C were: *minimal confidence* 0.8, *minimal support* 0.03 and *'use a weighting scheme to select best rules'* as the post-processing scheme. We used the version of RIPPER implemented in WEKA [17] with default parameters.

Furthermore we present also the results showing the comparison of APRIORI-SD with the algorithms CN2 and CN2-SD. Note that these results were obtained from the original (non-discretized) UCI data sets and are included to enable an additional comparison with a standard classification rule learner (CN2) and a state-of-the-art subgroup discovery algorithm (CN2-SD).

Table 3 presents summary results of the comparisons on UCI data sets, while details can be found in Tables 4, 5, 6 and 7 in the Appendix. For each performance measure, the summary table shows the average value over all the data sets, the significance of the results compared to APRIORI-SD (p-value), and the

Table 2. Data set characteristics.

	Data set	#Attr.	#Class.	#Ex.	Maj. Class (%)
1	australian	14	2	690	56
2	breast-w	9	2	699	66
3	bridges-td	7	2	102	85
4	chess	36	2	3196	52
5	diabetes	8	2	768	65
6	echo	6	2	131	67
7	german	20	2	1000	70
8	heart	13	2	270	56
9	hepatitis	19	2	155	79
10	hypothyroid	25	2	3163	95
11	ionosphere	34	2	351	64
12	iris	4	3	150	33
13	mutagen	59	2	188	66
14	mutagen-f	57	2	188	66
15	tic-tac-toe	9	2	958	65
16	vote	16	2	435	61
17	balance	4	3	625	46
18	car	6	4	1728	70
19	glass	9	6	214	36
20	image	19	7	2310	14
21	soya	35	19	683	13
22	waveform	21	3	5000	34
23	wine	13	3	178	40

Table 3. Summary of the experimental results comparing APRIORI-SD with different algorithms using 10-fold stratified cross-validation.

Performance Measure	Data Sets	**APRIORI-SD**	**APRIORI-C**	**RIPPER**	**CN2**	**CN2-SD**
Coverage	23	**0.53** ± 0.26	0.36 ± 0.19	0.19 ± 0.19	0.13 ± 0.14	0.46 ± 0.25
• sig. (p value)			0.000	0.000	0.000	0.012
• loss/win/draw			22/1/0	22/1/0	22/1/0	19/4/0
Size	23	**3.6** ± 1.96	5.6 ± 2.84	16.1 ± 27.47	18.2 ± 21.77	6.4 ± 4.58
• sig. (p value)			0.000	0.035	0.003	0.000
• loss/win/draw			21/2/0	20/3/0	22/1/0	17/4/2
Likelihood ratio	23	**12.37** ± 7.26	2.60 ± 0.55	2.36 ± 0.55	2.11 ± 0.46	18.47 ± 9.00
• sig. (p value)			0.000	0.000	0.000	0.000
• loss/win/draw			22/1/0	22/1/0	22/1/0	8/15/0
AUC	16	**82.80** ± 8.70	80.92 ± 9.95	80.11 ± 10.23	82.16 ± 16.81	86.33 ± 8.60
• sig. (p value)			0.190	0.027	0.871	0.011
• loss/win/draw			10/6/0	12/4/0	5/11/0	3/13/0
Accuracy	23	**79.98** ± 16.67	81.02 ± 16.50	83.46 ± 10.24	81.61 ± 11.66	79.36 ± 16.24
• sig. (p value)			0.039	0.282	0.489	0.414
• loss/win/draw			10/13/0	13/10/0	15/8/0	12/11/0

LOSS/WIN/DRAW in terms of the number of domains in which the results are worse/better/equal compared to APRIORI-SD. The analysis shows the following:

– In terms of the average coverage per rule APRIORI-SD produces rules with significantly higher coverage (higher the coverage better the rule) than both APRIORI-C and RIPPER. Moreover it produces rules with significantly higher coverage than both CN2 and CN2-SD.
– APRIORI-SD induces rule sets that are significantly smaller than those induced by APRIORI-C and RIPPER (smaller rule sets are better). It induces also significantly smaller rule sets than both CN2 and CN2-SD.

- APRIORI-SD induces significantly better rules in terms of significance (rules with higher significance are better) - computed by the average likelihood ratio - than APRIORI-C, RIPPER and CN2, while being significantly worse than CN2-SD. Note: APRIORI-SD, CN2 and CN2-SD rules are already significant at the 99% level.
- As the comparisons in terms of the area under the ROC curve (AUC) are restricted to binary class data sets, only the 16 binary data sets were used in this comparison. Notice that while being better than APRIORI-C and RIPPER, APRIORI-SD is comparable to CN2, but worse than CN2-SD.

Despite the fact that subgroup discovery is not intended at maximizing accuracy, it is worthwhile noticing that in terms of predictive accuracy APRIORI-SD is just insignificantly worse than the other three rule learners (APRIORI-C, RIPPER and CN2) and insignificantly better than CN2-SD.

5 Conclusions

Following the ideas presented in [10] we have adapted the APRIORI-C algorithm to subgroup discovery (APRIORI-SD). Experimental results on 23 UCI data sets demonstrate that APRIORI-SD produces smaller rule sets, where individual rules have higher coverage and significance compared to other rule learners (APRIORI-C, RIPPER and CN2). These three factors are important for subgroup discovery: smaller size enables better understanding, higher coverage means larger support, and higher significance means that rules describe discovered subgroups that are significantly different from the entire population.

When comparing APRIORI-SD to another subgroup discovery algorithm (CN2-SD) it turns out that APRIORI-SD still produces smaller rule sets, where individual rules have higher coverage, but they are less significant than the CN2-SD rules. Note that these results need further analysis as CN2-SD rules were induced from non-discretized data while APRIORI-SD used discretized data sets.

We have evaluated the results of APRIORI-SD also in terms of classification accuracy and AUC and shown a small increase in terms of the area under the ROC curve compared to APRIORI-C and RIPPER. A decrease in AUC compared to (CN2 and) CN2-SD could again be attributed to the use of non-discretized attributes in CN2(-SD).

In further work we plan to study the theoretical properties of the weighted covering approach. Our plans are also to compare our algorithm with the subgroup discovery algorithm MIDOS. Finally, we plan to use the APRIORI-SD subgroup discovery algorithm for solving practical problems, in which expert evaluations of induced subgroup descriptions is of ultimate interest.

Acknowledgements. The work reported in this paper was supported by the Slovenian Ministry of Education, Science and Sport.

References

1. R. Agrawal, T. Imielinski and R. Shrikant. Mining Association Rules between Sets of Items in Large Databases. In *Proceedings of ACM SIGMOD Conference on Management of Data*, Washington, D.C., 1993. pp. 207–216.
2. R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In *Proceedings of the 20th International Conference on Very Large Databases*, Santiago, Chile, 1994. pp. 207–216.
3. P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4): 261–283, 1989.
4. W.W. Cohen. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufmann, 1995. pp. 115–123.
5. S. Džeroski, N. Lavrač. (Eds). *Relational Data Mining*, Springer, 2001. pp. 74–99.
6. C. Ferri-Ramírez, P.A. Flach, and J. Hernandez-Orallo. Learning decision trees using the area under the ROC curve. In *Proceedings of the 19th International Conference on Machine Learning*, Morgan Kaufmann, 2002. pp. 139–146.
7. V. Jovanoski, N. Lavrač. Classification Rule Learning with APRIORI-C. In P. Brazdil, A. Jorge (Eds.): *10th Portuguese Conference on Artificial Intelligence*, EPIA 2001, Porto, Portugal, 2001. pp. 44–51.
8. I. Kononenko. On Biases in Estimating Multi-Valued Attributes. In *Proceedings of the 14th Int. Joint Conf. on Artificial Intelligence*, 1995. pp. 1034–1040.
9. N. Lavrač, P. Flach, and B. Zupan. Rule evaluation measures: A unifying view. In *Proceedings of the 9th International Workshop on Inductive Logic Programming*, Springer, 1999. pp. 74–185.
10. N. Lavrač, P. Flach, B. Kavšek, L. Todorovski. Adapting classification rule induction to subgroup discovery. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'02)*, Maebashi City, Japan, 2002. pp. 266–273.
11. B. Liu, W. Hsu and Y. Ma. Integrating Classification and Association Rule Mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining KDD'98*, New York, USA, 1998. pp. 80–86.
12. R.S. Michalski, I. Mozetič, J. Hong, and N. Lavrač. The multi-purpose incremental learning system AQ15 and its testing application on three medical domains. In *Proc. 5th National Conference on Artificial Intelligence*, Morgan Kaufmann, 1986. pp. 1041–1045.
13. P.M. Murphy and D.W. Aha. *UCI repository of machine learning databases* [http://www.ics.uci.edu/˜mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 1994.
14. F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3): 203–231, 2001.
15. R.L. Rivest. Learning decision lists. *Machine Learning*, 2(3): 229–246, 1987.
16. L. Todorovski, P. Flach, and N. Lavrač. Predictive performance of weighted relative accuracy. In D.A. Zighed, J. Komorowski, and J. Zytkow, editors, *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, Springer, 2000. pp. 255–264.
17. I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, 1999.
18. S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery*, Springer, 1997. pp. 78–87.

Appendix

Table 4. Average size (*S*), coverage (*CVG*) and likelihood ratio (*LHR*) of rules with standard deviations using 10-fold stratified cross-validation - algorithms: APRIORI-SD, APRIORI-C and RIPPER.

	APRIORI-SD			APRIORI-C			RIPPER		
#	*S* ± *sd*	*CVG* ± *sd*	*LHR* ± *sd*	*S* ± *sd*	*CVG* ± *sd*	*LHR* ± *sd*	*S* ± *sd*	*CVG* ± *sd*	*LHR* ± *sd*
1	3.5 ± 0.15	0.55 ± 0.06	8.40 ± 0.04	2.6 ± 0.51	0.43 ± 0.04	2.29 ± 0.03	11.6 ± 1.01	0.09 ± 0.01	2.77 ± 0.18
2	4.2 ± 0.43	0.30 ± 0.03	14.24 ± 0.02	8.0 ± 0.26	0.19 ± 0.02	3.08 ± 0.14	10.7 ± 0.12	0.10 ± 0.01	1.51 ± 0.55
3	2.4 ± 0.51	0.54 ± 0.05	8.16 ± 0.02	2.7 ± 0.04	0.60 ± 0.06	2.77 ± 0.05	1.4 ± 0.15	0.80 ± 0.08	3.30 ± 0.12
4	1.4 ± 0.20	0.53 ± 0.05	9.78 ± 0.15	3.2 ± 0.38	0.50 ± 0.05	2.87 ± 0.06	17.5 ± 0.83	0.05 ± 0.00	1.81 ± 0.05
5	4.4 ± 0.30	0.30 ± 0.03	16.40 ± 0.06	3.9 ± 0.35	0.28 ± 0.03	2.52 ± 0.07	10.2 ± 0.32	0.09 ± 0.01	2.31 ± 0.19
6	1.0 ± 0.00	1.00 ± 0.10	10.35 ± 0.03	3.5 ± 0.00	0.71 ± 0.07	2.37 ± 0.04	2.9 ± 0.12	0.39 ± 0.04	1.80 ± 0.75
7	6.2 ± 0.05	0.30 ± 0.03	10.98 ± 0.05	9.7 ± 0.83	0.11 ± 0.01	2.53 ± 0.02	11.5 ± 1.13	0.07 ± 0.01	2.57 ± 0.02
8	1.4 ± 0.14	0.67 ± 0.07	5.22 ± 0.06	4.4 ± 0.10	0.28 ± 0.03	2.69 ± 0.03	5.2 ± 0.04	0.16 ± 0.02	1.91 ± 0.08
9	2.8 ± 0.72	0.85 ± 0.08	10.53 ± 0.12	4.4 ± 0.28	0.52 ± 0.05	2.75 ± 0.06	2.5 ± 0.69	0.40 ± 0.04	2.76 ± 0.60
10	1.4 ± 0.71	0.52 ± 0.05	4.15 ± 0.04	3.0 ± 0.08	0.50 ± 0.05	1.83 ± 0.08	9.3 ± 0.14	0.10 ± 0.01	2.31 ± 0.65
11	3.5 ± 0.63	0.24 ± 0.02	1.86 ± 0.02	5.8 ± 0.13	0.22 ± 0.02	2.09 ± 0.03	6.6 ± 0.02	0.16 ± 0.02	1.81 ± 0.04
12	2.1 ± 0.49	0.84 ± 0.08	7.45 ± 0.03	2.5 ± 0.61	0.52 ± 0.05	2.86 ± 0.06	1.8 ± 0.75	0.52 ± 0.05	2.41 ± 0.04
13	2.8 ± 0.14	0.91 ± 0.09	15.25 ± 0.05	4.1 ± 0.07	0.48 ± 0.05	2.25 ± 0.18	3.3 ± 0.16	0.28 ± 0.03	2.48 ± 0.64
14	2.3 ± 0.27	0.88 ± 0.09	15.26 ± 0.03	4.0 ± 0.23	0.47 ± 0.05	3.43 ± 0.02	2.4 ± 0.05	0.23 ± 0.02	2.12 ± 0.13
15	7.1 ± 0.22	0.29 ± 0.03	15.18 ± 0.17	10.3 ± 1.00	0.13 ± 0.01	3.24 ± 0.04	25.9 ± 1.94	0.04 ± 0.00	3.33 ± 0.48
16	2.0 ± 0.44	0.71 ± 0.07	12.00 ± 0.03	4.2 ± 0.41	0.68 ± 0.07	3.33 ± 0.06	5.7 ± 0.55	0.19 ± 0.02	1.54 ± 0.02
17	4.2 ± 0.18	0.38 ± 0.04	5.61 ± 0.06	6.2 ± 0.55	0.23 ± 0.02	3.00 ± 0.04	24.0 ± 1.60	0.04 ± 0.00	3.05 ± 0.06
18	5.8 ± 0.56	0.26 ± 0.03	25.02 ± 0.07	6.8 ± 0.17	0.16 ± 0.02	2.31 ± 0.05	34.5 ± 3.01	0.03 ± 0.00	2.05 ± 0.33
19	2.8 ± 0.09	0.84 ± 0.08	2.53 ± 0.08	5.3 ± 0.45	0.37 ± 0.04	1.17 ± 0.06	5.9 ± 0.17	0.15 ± 0.02	2.97 ± 0.60
20	5.3 ± 0.20	0.24 ± 0.03	29.47 ± 0.19	9.7 ± 0.94	0.14 ± 0.01	2.22 ± 0.12	21.7 ± 1.34	0.04 ± 0.00	2.66 ± 0.66
21	8.2 ± 0.24	0.28 ± 0.03	16.78 ± 0.05	12.9 ± 1.12	0.10 ± 0.01	2.63 ± 0.04	17.2 ± 1.17	0.06 ± 0.01	1.54 ± 0.33
22	5.1 ± 0.15	0.24 ± 0.03	26.36 ± 0.18	7.3 ± 0.20	0.19 ± 0.02	3.52 ± 0.07	135.3 ± 12.7	0.01 ± 0.00	2.75 ± 0.07
23	2.4 ± 0.21	0.62 ± 0.06	13.52 ± 0.04	4.5 ± 0.36	0.55 ± 0.06	2.06 ± 0.02	3.4 ± 0.20	0.38 ± 0.04	2.52 ± 0.05
∅	3.6 ± 1.96	0.53 ± 0.26	12.37 ± 7.26	5.6 ± 2.83	0.36 ± 0.19	2.60 ± 0.55	16.1 ± 27.46	0.19 ± 0.19	2.36 ± 0.55

Table 5. Accuracy (*Acc*) and area under the ROC curve (*AUC*) of rules with standard deviations using 10-fold stratified cross-validation - algorithms: APRIORI-SD, APRIORI-C and RIPPER.

	APRIORI-SD		APRIORI-C		RIPPER	
#	*Acc* ± *sd*	*AUC* ± *sd*	*Acc* ± *sd*	*AUC* ± *sd*	*Acc* ± *sd*	*AUC* ± *sd*
1	87.26 ± 7.80	84.14 ± 2.00	89.99 ± 8.29	82.11 ± 2.01	82.03 ± 7.40	83.22 ± 7.80
2	97.48 ± 8.90	88.99 ± 3.05	95.85 ± 9.36	91.50 ± 3.11	94.76 ± 8.60	90.07 ± 8.02
3	86.02 ± 7.97	81.15 ± 2.03	87.17 ± 8.08	85.96 ± 2.03	86.17 ± 7.80	84.14 ± 7.59
4	96.16 ± 8.62	90.79 ± 3.02	94.52 ± 9.13	90.97 ± 3.00	98.90 ± 8.95	88.94 ± 7.91
5	75.00 ± 7.41	76.94 ± 4.06	74.95 ± 7.32	76.25 ± 4.09	71.29 ± 6.53	76.34 ± 7.41
6	67.90 ± 5.84	66.48 ± 1.05	71.66 ± 6.51	67.18 ± 1.08	67.85 ± 6.31	63.27 ± 5.69
7	69.52 ± 5.98	74.25 ± 4.25	71.19 ± 6.22	70.98 ± 4.08	72.52 ± 6.62	66.95 ± 6.66
8	79.83 ± 7.08	85.13 ± 2.07	79.57 ± 7.22	75.47 ± 2.09	69.88 ± 6.02	72.70 ± 6.84
9	82.30 ± 7.46	84.08 ± 3.06	82.44 ± 7.73	78.86 ± 3.03	81.36 ± 7.82	79.58 ± 7.82
10	99.91 ± 9.11	93.16 ± 4.00	99.20 ± 9.82	97.29 ± 4.00	99.16 ± 8.99	96.36 ± 9.48
11	88.97 ± 8.70	90.09 ± 2.08	92.44 ± 8.71	75.58 ± 2.04	86.34 ± 8.50	88.52 ± 8.28
12	95.19 ± 9.33	90.82 ± 2.00	95.59 ± 8.65	89.83 ± 2.01	96.01 ± 9.60	90.20 ± 8.99
13	79.40 ± 7.30	78.84 ± 3.10	81.72 ± 7.69	77.50 ± 3.12	76.51 ± 6.79	75.31 ± 7.25
14	79.17 ± 7.01	72.32 ± 3.08	81.08 ± 7.89	77.98 ± 3.00	74.21 ± 7.05	74.30 ± 6.98
15	75.21 ± 7.21	71.69 ± 3.68	80.15 ± 7.23	62.90 ± 4.02	85.79 ± 8.44	63.42 ± 6.29
16	96.47 ± 8.79	96.00 ± 1.06	94.63 ± 9.07	94.38 ± 1.05	93.47 ± 9.04	88.42 ± 8.43
17	75.13 ± 7.21		77.93 ± 6.86		79.48 ± 7.87	
18	85.21 ± 7.11		84.47 ± 7.64		92.73 ± 8.50	
19	66.49 ± 6.23		66.09 ± 6.05		64.84 ± 5.70	
20	19.98 ± 1.89		18.25 ± 1.37		86.42 ± 7.92	
21	68.21 ± 6.54		71.89 ± 7.17		89.39 ± 7.96	
22	75.58 ± 6.32		81.01 ± 7.47		78.85 ± 6.90	
23	93.23 ± 9.18		91.64 ± 8.29		91.51 ± 8.51	
Avg	79.98 ± 16.67	82.80 ± 8.70	81.02 ± 16.50	80.92 ± 9.95	83.46 ± 10.24	80.11 ± 10.23

Table 6. Average size (*S*), coverage (*CVG*), likelihood ratio (*LHR*), accuracy (*Acc*) and area under the ROC curve (*AUC*) of rules with standard deviations using 10-fold stratified cross-validation for the algorithm CN2.

	CN2				
#	*S* ± *sd*	*CVG* ± *sd*	*LHR* ± *sd*	*Acc* ± *sd*	*AUC* ± *sd*
1	12.4 ± 0.01	0.07 ± 0.01	2.0 ± 0.01	81.62 ± 0.01	33.39 ± 0.01
2	12.6 ± 0.10	0.08 ± 0.10	2.7 ± 0.10	92.28 ± 0.10	90.74 ± 0.10
3	1.8 ± 0.03	0.63 ± 0.03	2.1 ± 0.03	82.45 ± 0.03	84.51 ± 0.03
4	14.6 ± 0.01	0.05 ± 0.01	2.4 ± 0.01	94.18 ± 0.01	96.22 ± 0.01
5	12.8 ± 0.08	0.06 ± 0.08	2.0 ± 0.08	72.77 ± 0.08	71.33 ± 0.08
6	3.7 ± 0.06	0.31 ± 0.06	1.9 ± 0.06	68.71 ± 0.06	70.53 ± 0.06
7	15.1 ± 0.08	0.05 ± 0.08	2.0 ± 0.08	72.40 ± 0.08	71.99 ± 0.08
8	6.4 ± 0.09	0.11 ± 0.09	1.9 ± 0.09	74.10 ± 0.09	74.17 ± 0.09
9	3.0 ± 0.04	0.21 ± 0.04	2.7 ± 0.04	80.74 ± 0.04	78.81 ± 0.04
10	10.1 ± 0.00	0.09 ± 0.00	1.4 ± 0.00	98.58 ± 0.00	96.22 ± 0.00
11	7.6 ± 0.05	0.10 ± 0.05	2.0 ± 0.05	91.44 ± 0.05	94.46 ± 0.05
12	3.8 ± 0.01	0.38 ± 0.01	1.9 ± 0.01	91.33 ± 0.01	99.17 ± 0.01
13	4.7 ± 0.11	0.16 ± 0.11	2.1 ± 0.11	80.87 ± 0.11	83.20 ± 0.11
14	5.2 ± 0.01	0.14 ± 0.01	2.5 ± 0.01	72.28 ± 0.01	75.06 ± 0.01
15	21.2 ± 0.01	0.03 ± 0.01	2.5 ± 0.01	98.01 ± 0.01	97.90 ± 0.01
16	7.1 ± 0.01	0.13 ± 0.01	2.6 ± 0.01	94.24 ± 0.01	96.88 ± 0.01
17	28.7 ± 0.00	0.02 ± 0.00	2.7 ± 0.00	74.71 ± 0.00	
18	83.8 ± 0.05	0.02 ± 0.05	1.5 ± 0.05	89.82 ± 0.05	
19	12.9 ± 0.01	0.07 ± 0.01	1.0 ± 0.01	60.60 ± 0.01	
20	32.8 ± 0.11	0.04 ± 0.11	1.5 ± 0.11	58.88 ± 0.11	
21	35.1 ± 0.01	0.04 ± 0.01	2.4 ± 0.01	88.73 ± 0.01	
22	77.3 ± 0.01	0.00 ± 0.01	2.6 ± 0.01	69.18 ± 0.01	
23	5.5 ± 0.01	0.23 ± 0.01	2.0 ± 0.01	89.16 ± 0.01	
Avg	18.2 ± 21.77	0.13 ± 0.14	2.11 ± 0.46	81.61 ± 11.66	82.16 ± 16.81

Table 7. Average size (*S*), coverage (*CVG*), likelihood ratio (*LHR*), accuracy (*Acc*) and area under the ROC curve (*AUC*) of rules with standard deviations using 10-fold stratified cross-validation for the algorithm CN2-SD.

	CN2-SD (additive)				
#	*S* ± *sd*	*CVG* ± *sd*	*LHR* ± *sd*	*Acc* ± *sd*	*AUC* ± *sd*
1	3.5 ± 0.54	0.42 ± 0.03	4.6 ± 0.03	88.35 ± 0.03	85.12 ± 0.03
2	9.2 ± 0.25	0.26 ± 0.04	26.6 ± 0.04	92.60 ± 0.04	94.52 ± 0.04
3	1.8 ± 0.06	0.33 ± 0.02	22.9 ± 0.02	81.46 ± 0.02	83.03 ± 0.02
4	8.5 ± 0.32	0.51 ± 0.04	30.2 ± 0.04	96.08 ± 0.04	92.87 ± 0.04
5	4.6 ± 0.33	0.38 ± 0.04	2.1 ± 0.04	74.12 ± 0.04	80.06 ± 0.04
6	3.4 ± 0.05	0.87 ± 0.05	23.1 ± 0.05	68.75 ± 0.05	70.61 ± 0.05
7	8.8 ± 0.87	0.15 ± 0.12	16.3 ± 0.12	72.40 ± 0.12	72.73 ± 0.12
8	1.8 ± 0.18	0.44 ± 0.09	30.6 ± 0.09	78.03 ± 0.09	85.62 ± 0.09
9	2.7 ± 0.24	0.69 ± 0.03	25.0 ± 0.03	86.14 ± 0.03	81.29 ± 0.03
10	2.5 ± 0.03	0.51 ± 0.00	13.5 ± 0.00	99.10 ± 0.00	97.42 ± 0.00
11	4.2 ± 0.12	0.35 ± 0.06	14.9 ± 0.06	91.10 ± 0.06	93.87 ± 0.06
12	3.6 ± 0.68	0.67 ± 0.01	4.0 ± 0.01	91.75 ± 0.01	99.46 ± 0.01
13	4.5 ± 0.09	0.62 ± 0.17	9.7 ± 0.17	80.86 ± 0.17	83.06 ± 0.17
14	2.1 ± 0.27	0.72 ± 0.03	18.1 ± 0.03	77.74 ± 0.03	78.51 ± 0.03
15	10.2 ± 1.05	0.12 ± 0.03	26.5 ± 0.03	77.38 ± 0.03	89.15 ± 0.03
16	1.8 ± 0.43	0.83 ± 0.03	6.0 ± 0.03	97.62 ± 0.03	93.95 ± 0.03
17	8.3 ± 0.51	0.32 ± 0.00	24.3 ± 0.00	72.51 ± 0.00	
18	12.8 ± 0.17	0.20 ± 0.06	19.3 ± 0.06	72.48 ± 0.06	
19	10.1 ± 0.48	0.76 ± 0.01	9.1 ± 0.01	69.32 ± 0.01	
20	9.2 ± 0.99	0.17 ± 0.17	21.7 ± 0.17	19.49 ± 0.17	
21	19.2 ± 1.14	0.22 ± 0.03	30.6 ± 0.03	71.04 ± 0.03	
22	11.7 ± 0.23	0.19 ± 0.03	20.2 ± 0.03	75.70 ± 0.03	
23	1.4 ± 0.32	0.89 ± 0.03	25.7 ± 0.03	91.32 ± 0.03	
Avg	6.4 ± 4.60	0.46 ± 0.25	18.47 ± 9.00	79.36 ± 16.24	86.33 ± 8.60