Inducing Small and Accurate Decision Trees

Bernhard Pfahringer

Austrian Research Institute for Artificial Intelligence,
Schottengasse 3, A-1010 Vienna, Austria
E-mail: bernhard@ai.univie.ac.at
Phone: +43 1 533 6112 17

Keywords: Decision-tree learning, Pruning, Noise
Multiple submission statement: not applicable

Abstract

Recently, the quality improvement of decision trees and classifiers in general achievable by extended search efforts has received quite some attention in the literature. Contrary to the construction of ensembles of classifiers, which aims at improving overall predictive accuracy, our approach aims at improving the intelligibility of a single classifier. Our goal is the induction of a single, small, yet accurate decision tree. We describe a simple prepruning method (PreC4) that uses cross-validation to determine an appropriate stopping point for tree construction in a reliable manner. In addition to comparison with C4.5, PreC4 is also evaluated against both Robust-C4 and the combination of the two methods (Robust-PreC4). Evaluation domains comprise two artificial problems as well as a selection of small- and medium-sized UCI databases. Experimental results confirm that trees generated by both PreC4 and Robust-C4 are reasonably accurate but at the same time consistently smaller than trees generated by C4.5. PreC4 usually achieves a much larger tree-size reduction than Robust-C4 does. Interestingly, the combined procedure Robust-PreC4 does not perform as well. Trees generated by Robust-PreC4 are the smallest ones overall, but unfortunately they are also less accurate in some domains where they seemingly underfit the respective target concepts. In summary, PreC4 induces much smaller trees of comparable predictive accuracy.

1 Introduction

Recently, the quality improvement of decision trees and classifiers in general achievable by extended search efforts has received quite some attention in the literature. Contrary to the construction of ensembles of classifiers, which aims at improving overall predictive accuracy, our approach aims at improving the intelligibility of a single classifier. Our goal is the induction of a single, small, yet accurate decision tree.

It is common practice in decision tree learning to first grow a rather large, possibly over-fitting tree and then subsequently prune it back to some smaller size. This *post-pruning* process is usually guided by either statistical measures of significance (e.g. chi-square testing), or some form of error estimate (e.g. pessimistic error estimates [Quinlan 93], or reduced error pruning), or it might also take into account the complexity of the induced tree (e.g. minimum description length calculations as in [Quinlan & Rivest 89] or cost-complexity pruning used by the CART system [Breiman et al. 84]). The rationale for using post-pruning instead of pre-pruning is the hope to counter the detrimental effects of limited lookahead in the recursive construction of a tree. If the algorithm cannot find a good split locally, this might be due to either there being no useful way for further splitting or it might be due to interactions between attributes. Whereas in the former case immediately stopping would be the appropriate action to take, in the latter case continuing the building process may reveal some useful sub-tree. It is hoped that subsequent post-pruning will be able to distinguish between these two cases and remove all unnecessary sub-trees.

However, some recent studies shed doubt onto this hope. Standard post-pruning algorithms might not be effective or radical enough in pruning overly complex trees, as significantly simpler trees often exhibit comparable predictive accuracy. In [Holte 93, Auer et al. 95] very simple decision-trees (being restricted to one or two levels respectively) are shown to perform well in quite a few domains. In [Mehta et al. 95] a minimum description length based formula allows to post-prune trees considerably while retaining their predictive accuracy. [John 95] introduces ROBUST-C4 which iteratively discard misclassified examples, also resulting in smaller trees of similar predictive accuracy. Instead of discarding examples one can also discards a subset of the attributes used to describe the examples. This so-called feature subset selection [Cherkauer & Shavlik 96, Pfahringer 95] usually also leads to sig-

nificantly simpler trees of comparable or better predictive accuracy. Finally, in investigating the effect of training set sizes on decision tree complexity, [Oates & Jensen 97] finds a more or less linear relationship between these two factors for various post-pruning methods.

Pre-pruning is usually dismissed for being non-competitive. For instance, [Quinlan 93] reports that early experiments with a stopping criterion based on the chi-square test did not perform satisfactorily. Such static prepruning faces two problems: firstly, prespecified thresholds are more or less arbitrary, and secondly, hill-climbing search is inherently myopic. In the following we try to overcome these two problems by both estimating a reasonable threshold value from the given data itself by means of cross-validation, and by incorporating a slightly different, global tree-construction regime. In the next section we describe the new algorithm which we will call PREC4. Section 3 reports on experiments conducted in both artificial and natural domains. Section 4 concludes with a discussion of results and further research directions.

2 Pre-Pruning

Standard decision tree induction algorithms usually rely solely on some form of post-pruning to avoid overfitting the training set. For instance, C4.5[1] [Quinlan 93] uses a pessimistic error-estimate based on local error-rates, whereas so-called *cost-complexity pruning* as used in the CART system [Breiman et al. 84] accounts for both error-rates and the tree complexities by means of a weighted sum where the coefficients are estimated from the given data.

Contrary to these approaches, we try to further limit decision tree sizes during the tree growing phase already. In the following we will describe PREC4, which is a variant of C4.5 incorporating global error-based pre-pruning and a different tree-construction procedure. The key idea is trying to estimate, by means of cross-validation, a global error-rate that can reasonably be expected from a decision tree in some domain. Once an estimate is computed, a small tree with an error rate close to this estimate can be constructed.

[1] Actually C4.5 incorporates a crude form of pre-pruning, namely the "-m" parameter, which will be mentioned later.

Tree construction proceeds as follows: instead of the usual divide-and-conquer implementation a global heap of open search nodes is maintained where nodes which lead to a larger error-reduction are given a higher priority. Ties are broken in favor of nodes covering a larger number of examples. Using this regime it is easy stop induction once the total error on the training set has fallen below a given threshold value. Pseudo-code for this algorithm is depicted in Figure 1. The procedure *expand_open* constructs a proper decision tree node, splits up the training examples into the respective branches according to their attribute values, and for each such branch a search node recording the best next split-attribute and the respective error-reduction (as computed from the training-set) is enqueued into the heap. After reaching the target error rate, *finalize_open_heap_nodes* converts all search nodes still open into proper decision tree leaves that predict the respective majority class.

Function grow_tree
Input: Train set, Target error

1. **initialize**
2. $Heap := make\_node(Grow)$;
3. **repeat**
4. $Open := dequeue(Heap)$
5. $expand\_open(Open)$;
6. **until** $CurrentError \leq TargetError$
7. $finalize\_open\_heap\_nodes$;

Output: decision tree

Figure 1: Growing a tree up to a pre-specified target error

Now for computing an appropriate estimate of the target error, we only have to modify the above algorithm slightly. Instead of the target error an estimation set of examples distinct from the training set is supplied as input, and induction proceeds all the way down until the queue is empty. For a clear distinction we will call the training set the "growing" set. The returned estimate is the global error of this growing set for that iteration of induction where the minimal estimation set error was encountered for the first time.

The rationale for basing the estimate on the growing set instead of simply returning the minimum estimation set error is twofold. Firstly, estimation set sizes are much smaller, therefore this estimate has a larger variance rendering it less reliable, and secondly it might be overly optimistic, as it is the quantity being directly optimized by function *estimate_error*. On the other hand we cannot directly estimate the growing set error, as usually the growing set error can be driven close to zero by overfitting. Choosing the *first* occurrence of the minimal estimation set error for determining the appropriate growing set error is the natural way of enforcing the small-tree bias we are focussed on. The modified algorithm is depicted in Figure 2.

Function estimate_error
Input: Grow set, Estimation set

```
1.  initialize
2.      BestGrowError := MaxInt;
3.      BestEstError := MaxInt;
4.      Heap := make_node(Grow);
5.  while Open := dequeue(Heap) do
6.      expand_open(Open);
7.      if CurrentEstError < BestEstError
8.              then BestEstError := CurrentEstError;
9.                   BestGrowError := CurrentGrowError;
10. return BestGrowError
```

Output: Error Estimate

Figure 2: Estimating a reasonable target error rate

After unsatisfactory initial experiments which simply used the number of errors as an error estimate, we chose to incorporate a more sophisticated estimator. If we interpret each class-value in an n-class problem as a point in a corner of an n-dimensional hyper-cube of edge-length one, we can compute a mean class value for any set of examples, and the "sum-squared-error" for the set with respect to its mean. Not surprisingly, this sum-squared-error measure proved to be a good estimator, as it is equivalent to a geometric interpretation of the Gini-Index measure used in the CART system

[Breiman et al. 84] for choosing the best attribute to split on.

Estimating the final target sum-squared-error is done by averaging the target values computed by one ten-fold stratified cross-validation run of the estimation procedure on the training set, each time using 9 $(k-1)$ folds as the growing set and the remaining fold as the estimation set. As the sum-squared-error is of course dependent on the total number of examples, we have to be careful to account for different set-sizes correctly.

Finally we depict the high-level description of PREC4 in Figure 3. In addition to the target error estimation and subsequent tree construction we have also added a post-pruning phase equivalent to the post-pruning procedure used by C4.5, as we found that such additional post-pruning of already pre-pruned decision trees further reduced their size without impairing their predictive accuracy. Post-pruning is especially helpful in cases where the target-error is almost reached quite a few iterations before the final termination of tree-construction. Such a situation probably indicates too low target error estimates.

Algorithm PREC4
Input: Training set

1. $Target := average(cross\_val(estimate\_error, Training))$;
2. $Tree := grow\_tree(Training, Target)$;
3. **return** $postprune(Tree)$

Output: Decision-tree

Figure 3: The PREC4 algorithm

3 Empirical Evaluation

In this section we report on empirical results for two artificial domains and for a few standard benchmark domains as found in the UC Irvine repository [Merz & Murphy 96]. All the results given are averaged over ten runs of stratified ten-fold cross-validation. We always compare both predictive error-rates and final tree sizes for all four methods C4.5, ROBUST-C4 [John 95], PREC4, and ROBUST-PREC4. The size of a decision tree is defined to be

the number of tests in a tree. C4.5 uses default settings for its parameters: pruning confidence was set 25% (parameter -c25) and every test must have at least two branches with two or more examples covered (parameter -m2). ROBUST-C4 has already been described briefly in Section two, it too uses C4.5's default settings. PREC4's post-pruning phase uses the same pruning confidence value of 25%. Finally, ROBUST-PREC4 is just the straightforward application of ROBUST-C4's principle to the PREC4 algorithm. So ROBUST-PREC4 iteratively applies PREC4 onto the current training set, deletes all mis-classified examples from this training set and repeats this process until a fix-point is reached, i.e. until the training set does not shrink any further. The decision tree of this last iteration is returned by ROBUST-PREC4 as the final result.

3.1 Artificial domains

Artificial domains serve a good purpose, as they allow for the investigation of specific properties in an otherwise pure and controlled setting. Additionally one usually can easily vary training set sizes at will. In both artificial domains we have done experiments for training set sizes of 100, 500, 1000, 2000, 3000, 4000, 5000, and 10000 to illustrate effects due to a growing number of examples available for induction.

3.1.1 Random data

This domain comprises solely random data, testing the ability of algorithms to distinguish between chance correlations and actual patterns. Each example is described by 30 random boolean attributes and a boolean class variable. This class variable is "true" for approximately 80% of the data and "false" for the rest. As can be seen in Table 1 and Figure 4, all methods converge at a 20% error rate when supplied with a sufficient number of examples (where "sufficient" varies between 500 and 5000), but with qualitatively quite different trees. Whereas C4.5 and ROBUST-C4 induce rather bushy trees, both PREC4 and ROBUST-PREC4 return a correct single-leaf tree when given 1000 examples or more. But interestingly, first of all the bushy trees do not seem to impair C4.5's predictive accuracy, at least not when given enough examples. And secondly, tree-sizes for C4.5 too, get smaller with larger numbers of examples, and may even converge to 0, when given vastly more

examples as indicated by the tree-sizes achieved at the 10000 examples point.

Examples	C4.5	R-C4	PreC4	R-PreC4
100	29.60	28.70	22.20	23.10
500	25.32	23.12	20.06	20.04
1000	23.34	21.16	19.78	19.78
2000	21.45	20.22	19.58	19.58
3000	21.34	20.17	19.66	19.66
4000	21.10	20.06	19.82	19.82
5000	20.28	19.85	19.75	19.75
10000	19.92	19.86	19.86	19.86

Table 1: Averages error rates for Random data.

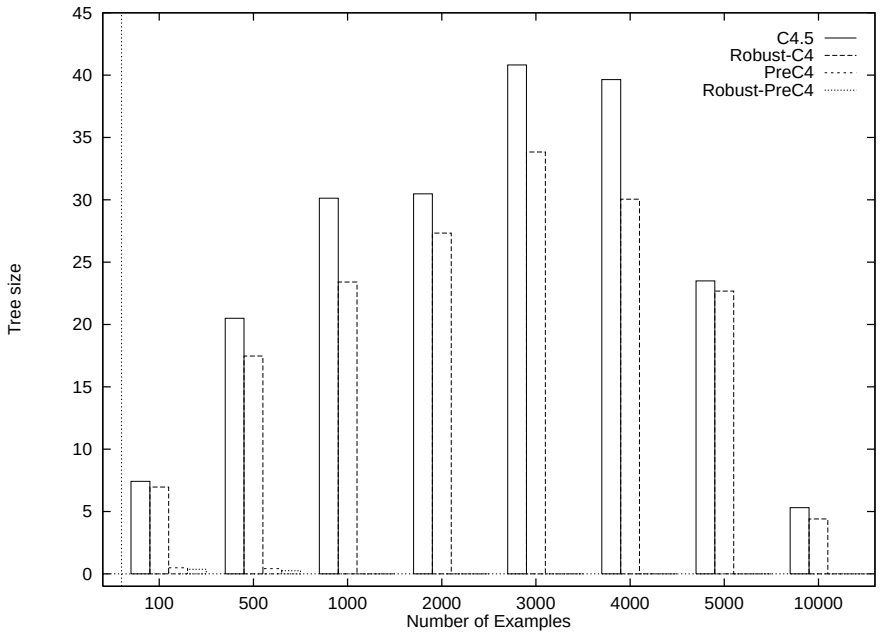

Figure 4: Random: Average tree size

3.1.2 Jensen

This domain was introduced in [Jensen 97] for evaluating the influence of the training set size on the pruning of decision trees. Once again, examples comprise 30 random boolean attributes and a boolean class variable. The class itself is computed by a decision tree of 5 tests on the first five boolean attributes. Additionally, the class variable is corrupted by class noise at a 10% level. That means that the class-label of 10% of the training examples was switched from "true" to "false" or vice versa. So the best one could hope for as an induction result would be the correct decision tree of size 5 and an error rate of 10%.

As shown in Table 2 and Figure 5, all methods quickly converge to optimal predictive accuracy, but C4.5 and ROBUST-C4 need a lot more training examples to also uncover the correct decision-tree. But we also note that for the smallest training set size of only 100 examples both PREC4 and ROBUST-PREC4 return trees which are significantly worse predictors (as judged by a paired t-test).

Additionally, it is interesting to note that the figures in [Jensen 97] seem to indicate monotonic growth of decision-tree sizes for their implementation of post-pruning based on pessimistic error estimates (that curve should be equivalent to C4.5 in Table 2). But this finding is probably misleading, because they have only used training sets of up to 250 examples. Given enough examples, all methods including C4.5 are able to induce the correct decision tree in this domain.

3.2 Natural Domains

To further test the influence of pre-pruning on tree sizes and accuracies, we also conducted experiments on a few database mostly available from the UCI repository [Merz & Murphy 96]. The specific databases were chosen to comprise a good mix of the various properties distinguishing flat-file databases: number of cases, number of classes, the number of categorical attributes, and the number of numerical attributes. Table 3 summarizes these coordinates for all databases used.

Average error rates for all domains are given in Table 4, and average tree-sizes are compared in Figure 6. Most differences in predictive accuracy are absolutely insignificant when judged by a paired t-test. The significant ones

Examples	C4.5	R-C4	PreC4	R-PreC4
100	20.80	20.30	22.2	24.2
500	12.30	11.96	10.68	10.62
1000	10.68	10.50	10.04	9.98
2000	9.98	9.96	9.86	9.85
3000	9.99	9.99	9.93	9.93
4000	9.95	9.95	9.94	9.94
5000	9.90	9.90	9.90	9.90
10000	9.98	9.97	9.97	9.97

Table 2: Averages error rates for Jensen data.

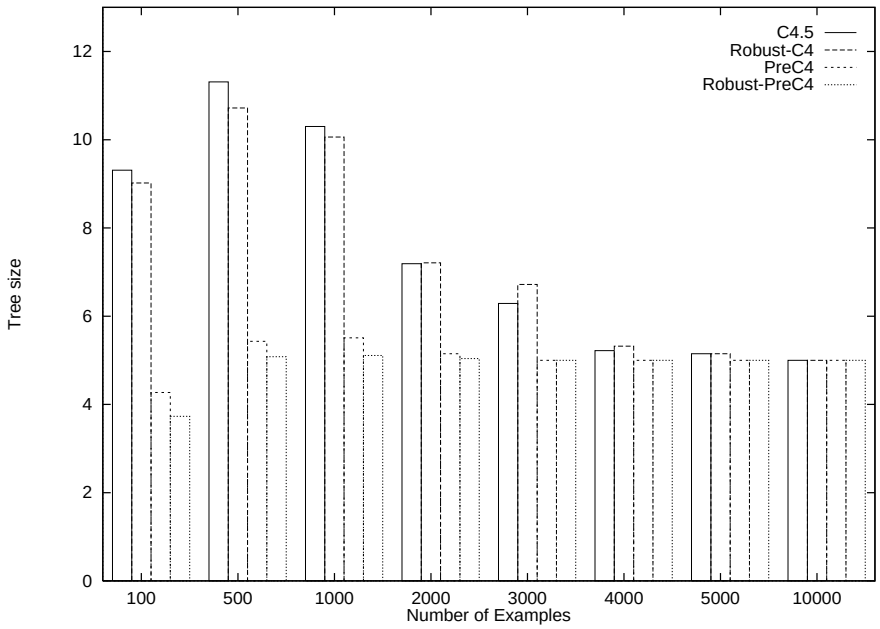

Figure 5: Jensen: Average tree size.

Domain	Cases	Classes	Cat	Num
BC: Breast-w	699	2	0	9
CR: Credit	690	2	9	6
CO: Colic	368	2	14	8
DI: Diabetes	768	2	0	8
DN: DNA	3186	3	60	0
GE: German	1000	2	13	7
GL: Glass	214	6	0	9
IR: Iris	150	3	0	4
LA: Labor	57	2	8	8
LY: Lymph	148	4	18	0
SO: Soybean	683	19	35	0
VO: Vote	435	2	16	0

Table 3: Domain properties.

are: in the LABOR domain and in the LYMPH both PREC4 and ROBUST-PREC4 perform worse. At least in the LABOR domain this effect is probably caused by the very small number of training examples available, which presumably leads the internal cross-validation estimator astray. For a pattern to be discovered enough examples showing this pattern must be present in both the growing and the estimation set, otherwise the pattern will either not be spotted at all or discarded later. The same might be true for the LYMPH domain as well, which comprises a few more examples, but also four different classes and also a larger attribute space. This effect was present for both artificial data sets described above as well. ROBUST-PREC4 does worse for both the IRIS and the SOYBEAN domain. In the SOYBEAN domain PREC4 achieves better predictive accuracy than all other methods apart from constructing a smaller decision-tree.

Comparing tree-sizes, we clearly see the expected relationship:

$$tree\_size_{C4.5} \geq tree\_size_{R-C4.5} \geq tree\_size_{PreC4} \geq tree\_size_{R-PreC4}$$

For the five domains BREAST, CREDIT, DIABETES, GERMAN, and VOTE radically smaller trees are induced by both PREC4 and ROBUST-PREC4.

Examples	C4.5	R-C4	PreC4	R-PreC4
Breast	5.97	5.69	5.83	5.95
Credit	15.29	15.04	15.62	15.41
Colic	15.65	15.63	14.76	14.43
Diabetes	25.86	25.95	25.85	26.18
DNA	6.42	6.49	6.35	6.49
Glass	35.19	34.44	34.58	34.81
German	27.45	27.54	27.79	27.48
Iris	5.87	5.87	5.87	6.6
Labor	18.95	18.95	23.68	23.33
Lymph	21.82	22.36	23.92	23.45
Soybean	8.04	8.33	7.34	9.34
Vote	4.74	4.83	4.62	4.62

Table 4: Averages error rates for several UCI databases.

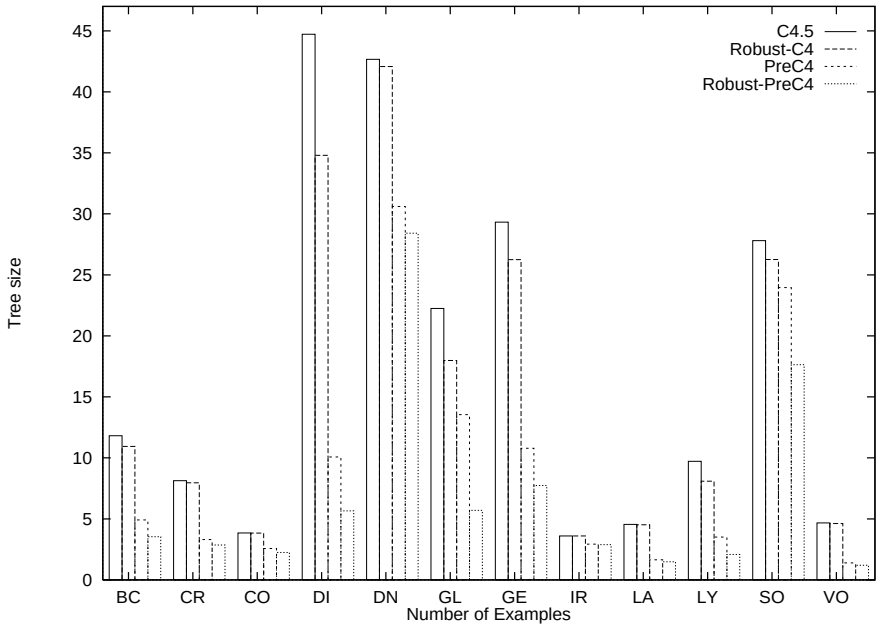

Figure 6: Tree sizes for several UCI databases.

To summarize all experimental findings, the extra computational effort (a constant factor N in runtime when using N-fold cross-validation) invested seems to pay off in most domains, sometimes yielding considerably smaller decision trees and also of comparable predictive accuracy. The only exception seem to be domains with very few training examples. Possibly, a different cross-validation regime could remedy this fault, e.g. using more than 10 folds or even leave-one-out cross-validation for such small training sets.

In general, these findings are quite reassuring given the recent discovery of the sometimes detrimental effects of so-called *oversearching* [Quinlan & Cameron-Jones 95, Murthy & Salzberg 95]. Another interesting conclusion is the fact that at least with respect to predictive accuracy C4.5's default settings seem to do very well for natural domains. Even the sometimes obvious overfitting apparently causes no or only small losses in predictive performance. But in some domains far less complex trees of comparable accuracy can vastly improve intelligibility.

4 Discussion & Conclusions

We have described a new simple but effective pre-pruning method for decision tree induction. PREC4 aims at inducing small trees of reasonable predictive accuracy. The experimental results reported in the previous section confirm PREC4's abilities. Further research directions include the following points:

- Some design decisions can be questioned, for instance the choice of sum-squared-error as an error-estimate. A reasonable alternative might be some form of information-gain or an MDL-based coding length measure. Also the choice of one 10-fold cross-validation run for estimating the target error is rather ad-hoc. Such decisions should be investigated in more detail.

- As has already been proven by both the 1R [Holte 93] and T2 [Auer et al. 95] systems, small decision trees can exhibit good predictive accuracy. But we do believe that PREC4's more dynamic bias which in a sense guesses the right decision tree size from the given data will be more appropriate in a general learner. This hypothesis should of course be evaluated in a separate study.

- Also a closer comparison to pruning based on so-called "intermediate decision trees" (IDT) as presented in [Holder 95] should be valuable. They try to directly estimate the right size for a decision tree also by means of cross-validation and a breadth-first tree-construction schema. But they report that C4.5 pruning outperforms IDT pruning in a majority of databases and leave necessary improvements to further research.

- Among others, decision lists and propositional rule-sets are important types of classifiers. Even though they are quite similar to decision trees, it is not immediately clear how pre-pruning in the manner of PREC4 could be incorporated into their usual separate-and-conquer regime.

- Current induction methodology makes good use of additional search effort to either increase predictive accuracy or intelligibility. Unfortunately, none of these methods is able to simultaneously optimize both criteria. Boosting PREC4 with only a small number of iterations and transforming the resulting ensemble into a single decision tree (maybe similar to work described in [Pfahringer 97]). might be an interesting starting point for such a synthesis.

Furthermore, we plan to adapt the key ideas of PREC4 to learning in a first-order framework (like [Blockeel & DeRaedt 97] or [Kramer 96]), where due to vastly larger concept spaces effective pruning is even more essential than it already is in a propositional setting.

Acknowledgements

The first idea for PREC4 was conceived while the author was visiting the Computer Science Department of the University of Waikato, Hamilton, New Zealand. The Austrian Research Institute for Artificial Intelligence is sponsored by the Austrian Federal Ministry of Science and Transport. I like to thank both Stefan Kramer and Gerhard Widmer for valuable discussion and for help in preparing this paper.

References

[Auer et al. 95] Auer P., Maass W., Holte R.: Theory and Applications of Agnostic PAC-Learning with Small Decision Trees, in Prieditis A. and Russell S.(eds.), *Proceedings of the 12th International Conference on Machine Learning (ML95)*, Morgan Kaufmann, San Francisco, 1995.

[Blockeel & DeRaedt 97] Blockeel H., DeRaedt L.: Top-down induction of logical decision trees, Technical Report Report CW 247, Katholieke Universiteit Leuven, Belgium, 1997.

[Breiman et al. 84] Breiman L., Friedman J.H., Olshen R.A., Stone C.J.: *Classification and Regression Trees*, Wadsworth International Group, Belmont, CA, The Wadsworth Statistics/Probability Series, 1984.

[Cherkauer & Shavlik 96] Cherkauer K.J., Shavlik J.W.: Growing Simpler Decision Trees to Facilitate Knowledge Discovery, in Simoudis E. and Han J.(eds.), *KDD-96: Proceedings Second International Conference on Knowledge Discovery & Data Mining*, AAAI Press, Menlo Park, pp.315-318, 1996.

[Holder 95] Holder L.B.: Intermediate Decision Trees, in Mellish C.S.(ed.), Proceedings of the 14th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Mateo, CA, pp.1056-1062, 1995.

[Holte 93] Holte R.C.: Very Simple Classification Rules Perform Well on Most Commonly Used Datasets, *Machine Learning*, 11(1), 1993.

[Jensen 97] Jensen D.: Adjusting for Multiple Testing in Decision Tree Pruning, Poster presentation at the *Sixth International Workshop on Artificial Intelligence and Statistics*, 1997.

[John 95] John G.H.: Robust Decision Trees: Removing Outliers from Databases, in Fayyad U.M. and Uthurusamy R.(eds.), *KDD-95: Proceedings First International Conference on Knowledge Discovery & Data Mining*, AAAI Press, Menlo Park, pp.174-179, 1995.

[Kramer 96] Kramer S.: Structural Regression Trees, in *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, AAAI Press/MIT Press, Cambridge, MA, pp.812-819, 1996.

[Mehta et al. 95] Mehta M., Rissanen J., Agrawal R.: MDL-Based Decision Tree Pruning, in Fayyad U.M. and Uthurusamy R.(eds.), *KDD-95: Proceedings First International Conference on Knowledge Discovery & Data Mining*, AAAI Press, Menlo Park, pp.216-221, 1995.

[Merz & Murphy 96] Merz C.J., Murphy P.M.: UCI Repository of machine learning databases Irvine, CA: University of California, Department of Information and Computer Science, 1996. [http://www.ics.uci.edu/ mlearn/MLRepository.html]

[Murthy & Salzberg 95] Murthy S.K., Salzberg S.: Lookahead and Pathology in Decision Tree Induction, in Mellish C.S.(ed.), *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, pp.1025-1031, 1995.

[Oates & Jensen 97] Oates T., Jensen D.: The Effects of Training Set Size on Decision Tree Complexity, in *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, 1997.

[Pfahringer 95] Pfahringer B.: Compression-Based Feature Subset Selection, in Turney P.(ed.), *IJCAI-95 Workshop on Data Engineering for Inductive Learning*, IJCAI-95 Workshop Program Working Notes, Montreal, Canada, 1995.

[Pfahringer 97] Pfahringer B.: On the Induction of Intelligible Ensembles, Oesterreichisches Forschungsinstitut fuer Artificial Intelligence, Wien, TR-97-30, 1997.

[Quinlan & Rivest 89] Quinlan J.R, Rivest R.L.: Inferring Decision Trees using the Minimum Description Length Principle, in Information and Computation, 80:227-248, 1989.

[Quinlan 93] Quinlan J.R.: C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.

[Quinlan & Cameron-Jones 95] Quinlan J., Cameron-Jones R.: Oversearching and Layered Search in Empirical Learning, in Mellish C.S.(ed.), *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, pp.1019-1024, 1995.