Intermediate Decision Trees

Lawrence B Holder
Department of Computer Science and Engineering
University of Texas at Arlington
Box 19015, Arlington, TX 76019-0015

Abstract

Intermediate decision trees are the subtrees of the full (unpruned) decision tree generated in a breadth-first order An extensive empirical investigation evaluates the classification error of intermediate decision trees and compares their performance to full and pruned trees Empirical results were generated using *C4* 5 with 66 databases from the UCI machine learning database repository Results show that when attempting to minimize the error of the pruned tree produced by C4 5, the best intermediate tree performs significantly better in 46 of the 66 databases These and other results question the effectiveness of decision tree pruning strategies and suggest further consideration of the full tree and its intermediates Also, the results reveal specific properties satisfied by databases in which the intermediate full tree performs best Such relationships improve guidelines for selecting appropriate inductive strategies based on domain properties

1 Introduction

Numerous decision, tree pruning methods have been developed to reduce decision tree error due to overfitting on the part of the decision tree induction method [Mingers, 1989] Of course, as Schaffer [1993, 1994] emphasizes no one pruning method can improve the performance of decision tree induction on all domains However, certain pruning methods can improve performance on certain domains Although this paper presents results indicating the possibility of yet another decision tree pruning method, the results also indicate in which domains such a pruning method might be useful

The perspective on which these results are based is the intermediate decision tree Section 3 defines the intermediate decision tree, but, in general, an intermediate decision tree is one of the sequence of subtrees generated using a breadth-first traversal of a full decision tree (l e , a decision tree whose leaf nodes contain examples of one class) Previous results have shown that the intermediate decision tree compares favorably to other pruning methods [Holder, 1992a], and intermediate learned concepts in general address overfitting issues in both inductive and speedup learning [Holder, 1990, 1992b, Holder and Chaudhry, 1993] The main result of this paper is that the best intermediate tree of a full decision tree is often better (less error) than the pruned tree The suggested pruning strategy would be to find the number of splits leading lo this minimum-error intermediate decision tree A pre-pruning strategy of performing just that number of splits would produce a pruned decision tree without the need to first generate the full tree

At one extreme to this approach, Holte [1993] suggests that the smallest intermediate decision tree consisting of one split performs well in comparison lo a larger decision tree However, Elomaa [1994] qualifies Holte's results by showing the small difference between C4 5 trees and one-split trees is significant in favor of the C 4 0 trees Elomaa further points out that in some domains simple classifiers will never outperform a multilevel decision tree Exhaustive tests run by Murphy and Pazzam [1994] indicate that the smallest consistent decision trees have more error than slightly larger consistent trees Weiss and Indurkhya [1994] conclude that with at least 100 examples their cross-validation cost-complexity pruning method outperforms no pruning Thus, although pruned decision trees clearly compare favorable to full decision trees, there is no indication that a particular level of tree complexity will prevail

The results presented in this paper deal exclusively with decision trees produced by the G4 5 program [Quinlan 1992], therefore, Section 2 briefly summarizes the properties of this program Section 3 discusses the intermediate decision tree in detail Section 4 presents the experimental results based on a large sampling of the University of California, Irvine Machine Learning Database Repository [Murphy and Aha, 1994] The appendices describe these databases and show a sample of the ex penmental results Section 5 considers specific hypotheses regarding the performance of intermediate decision trees as compared to C4 5 pruned trees, derives specific conditions underwhich intermediate decision trees perform best, and describes the implied pruning strategy Section 6 concludes with directions for future work

2 Decision Tree Induction

The method of decision tree induction used throughout this paper follows the method employed by C4 5 [Quinlan, 1992] C4 5 uses the standard recursive splitting technique to produce a decision tree whose leaf nodes contain training examples of one class If a split branch yields a node with no training examples, then this node is replaced by a leaf node whose classification is the majority class of the parent node If a node with examples from more than one class cannot be split further, then this node is replaced by a leaf node whose classification is the majority class of the examples at this node

C4 5 uses the gain ratio criterion for selecting the test attribute at each non-leaf node A continuous attribute *A* is split based on the best test $A < T$, where T is one of the values appearing in the examples Examples with unknown values for an attribute are distributed across all values and weighted according to the frequency of the known values in the examples Classification of examples with unknown values is treated similarly, where the class is based on the maximum sum of the weights of the classes in the leaf nodes reached by the example Although C4 5 is capable of splitting based on subsets of attribute values, this feature was not utilized in the experiments

C4 5 employs two tjpes of pruning During tree construction C4 5 requires that an\ split result in at least two branches having a minimum number weight *MW* of examples The default value of *MW* is 2, but can be changed by a program option Larger values of MW help prevent overfitting of noisy data The main post pruning strategy used in C4 5 is a form of pessimistic pruning in which a subtree s error is estimated based on the binomial probability distribution of *E* errors occurring within *N* trials with a confidence factor *C F* The default value of *CF* is 0 25, but can be changed by a program option Smaller values of *CF* tighten the error estimate and increase the amount of pruning In our experiments the values of *MW* and *CF* are tuned automatically using a hill climbing approach described in Section 4

3 Intermediate Decision Trees

An *intermediate decision tree* (IDT) of a decision tree is any subtree in the sequence of subtrees generated from a breadth-first traversal of the internal nodes (splits) of the decision tree For example, Figure la-d depicts the ordered sequence of IDTs for the decision tree in Figure le

Figure 2 reveals the motivation for the breadth-first traversal The figure depicts typical error curves that plot the error of different sequences of IDTs for a decision tree whose final error corresponds to the rightmost point on the curve[x] The best-first traversal orders the splits based on information gain Both the depth-first and best-first traversals indicate high error until a majority of the splits have been performed The breadth-first traversal, however, quickly achieves a low error, which

'These particular curves come from the DNF2 domain in [Pagallo and Haussler, 1990]

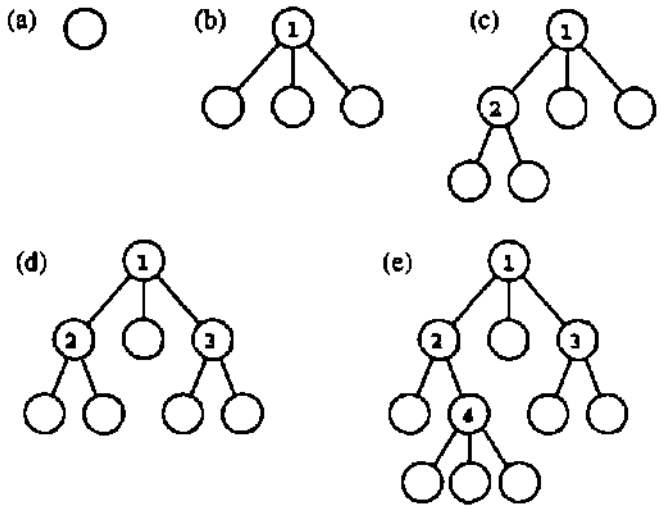

Figure 1 The decision trees in (a)-(d) are the intermediate decision trees of the decision tree in (e), which contains four splits

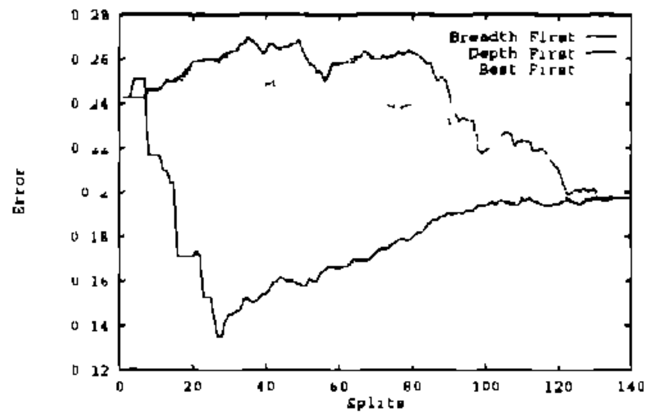

Figure 2 Three sequences of intermediate decision trees based on depth-first, breadth-first and best-first traversals of a decision tree

gradually ascends to the final error level This behavior of the trror curves is tvpical for many domains The specific behavior in which the minimum of the breadth-first traversal is less than the final error is the motivation for our interest in intermediate decision trees

4 Experiments

In order to evaluate the performance of intermediate trees, an extensive empirical investigation was performed using the decision tree induction program C4 5 Release 6 0 [Quinlan 1992] on 66 databases from the UCI Machine Learning Databases Repository [Murphy and Aha, 1994] Appendix A describes the databases

Four different trees were considered for comparison the full tree FT (no post pruning), the pruned tree PT, the best (lowest error) intermediate tree of the full tree IFT, and the best intermediate tree of the pruned tree

IPT Experiments measured the classification error and number of splits for each tree averaged over 50 trials each using two-thirds training and one-third testing examples randomly sampled without replacement Four different experiments were run, each attempting to minimize the error for a different tree

A hill-climbing parameter tuner accomplished the minimization by tuning the MW and CF parameters in order to minimize the average test-set error for one of the four tree types Starting from the default values ($MW = 2$, $CF = 0\ 25$), the tuner measures the average error (over 50 trials) for $MW \pm 1$ and $CF \pm 0\ 05$ If none of the four new parameter settings improve the error, then the search terminates Otherwise, the search continues from the best new parameter setting The resulting optimized trees are subsequently referred to as $\mathbf{FT}_{opt}$, $\mathbf{PT}_{opt}$, $\mathbf{IFT}_{opt}$ and $\mathbf{IPT}_{opt}$ Although the opli mized trees consider the test set to tune *MW* and *CF* these trees are used only to indicate what is possible The test set is not considered during the generation of the other trees or during the pruning phase

In addition to measuring the average error and number of splits for each of the four tree types, the experiment also measured the difference between the error and number of splits for the different pairwise combinations of trees A one-sided, upper-tailed, matched-pairs test measured the significance of the differences Subsequent mention of "significantly less' or "significantly greater implies a significance level of 0 05 or less (i e , the prob ability that the difference is significant is at least 0 95)

The results are too numerous to completely tabulate in this paper, but Appendix B tabulates a sample of the experimental results when optimizing the pruned tre

5 Discussion

Many conclusions can be drawn from the experimental results in terms of the performance of intermediate decision trees, the types of databases for which they are useful, and the trees' role in a pruning strategy

5 1 Hypotheses

Following is a series of hypotheses drawn from the experimental results The evidence indicates that intermediate decision trees outperform pruned trees in a majority of the databases

Hypothesis 1 1FT has less error than PT
Evidence 1 The results shown in Appendix D indicate that IFT has significantly less error than PT_{op}, in 46/66 databases PTpi has significantly less error than IFT in 5/66 databases Results not shown indicate that $IFT_{op(}$ has significantly less error than PT in 52/66 databases, and PT has significantly less error than $IFTo_{p(}$ in 1/66 databases

Hypothesis 2 IFT has less error than IPT
Evidence 2 IFT has significantly less error than the IPTop(in 20/66 databases IPTp, has significantly less error than IFT in 12/66 databases IFTppt has significantly leas error than IPT in 32/66 databases IPT has significantly less error than IFT in 7/66 databases

Hypothesis 3 IPT has less error than FT
Evidence 3 IPT has significantly less error than FT_0pt in 40/66 databases $\mathbf{FT}_{opt}$ has significantly less error than IPT in 5/66 databases $\mathbf{IPT}_{opt}$ has significantly less error than FT in 49/66 databases FT has significantly less error than $IPT_{op(}$ in 4/66 databases

Hypothesis 4 PT has less error than FT
Evidence 4 PT has significantly less error than FT_{op}i 27/66 databases $\mathbf{FT}_{opt}$ has significantly less error than PT in 10/66 databases $\mathbf{PT}_{opt}$ has significantly less error than FT in 32/66 databases FT has significantly less error than $\mathbf{PT}_{opt}$ in 7/66 databases

Hypothesis 5 IFT has less error than FT
Evidence 5 IFT has significantly less error than F $r_{op(}$ in 54/66 databases IFTopi has significantly less error than FT in 58/66 databases IFT never has greater error than FT

Hypothesis 6 IPT has less error than PT
Evidence 6 IPT has significantly less error than PT_{opr} in 53/66 databases IPT_{0prt} has significantly less error than PT in 53/66 databases IPT never has greater error than PT

Based on the evidence, IFT performs better than PT in a majority of the databases, indicating that intermediate trees deserve attention when attempting to minimize the error of a decision tree Furthermore, IPT performs better than PT in a majority of the databases, indicating that the pessimistic post-pruning used in C45 still has room for improvement IFT tends to outperform IPT indicting that a intermediate tree-based pruning strategy should consider intermediate trees of the full (unpruned) tree instead of intermediates of the pruned tree Also, the superiority of IFT over FT indicates that the breadth first ordering is capturing a minimum in the error curve less than the error for both full and pruned trees

Similar results measuring the number of splits indicate that the IFT has significantly fewer splits than FT, but significantly more splits than IPT Results comparing the number of splits between IFT and PT are inconclusive Thus although IFT has less error than PT and IPT this gain comes with an increased number of splits

5 2 Database Properties

Due to the large number of databases included in this study, we can attempt to discern patterns in the properties of databases in which the intermediate decision tree performs best Specifically, we used C4 5 to produce rules governing the database properties from Ta ble 2 after labeling each database as a positive or negative example of the hypothesis Specifically, databases in which IFT had significantly less error than $PT_{op(}$ were labeled as positive examples, the remaining databases were labeled negative Using these examples, the C4 5 rule generator [Quinlan, 1992] produced the following rules (the numbers to the right of each rule indicate the correct/incorrect classifications)

(Classes > 3) & (Cont = 0) → - [3/0]
(Size ≤ 23) → - [5/1]
(Classes ≤ 2) & (Cont = 0) & (A2 ≤ 2) → - [4/1]
(Size > 3197) → - [7/3]

(Size > 23) & (Error > 0 027) rightarrow + [41/1]
Default → +

These rules correctly classify 60/66 databases, where the errors consisted of one false positive and five false negatives The last rule, describing 42/66 databases suggests that IFT has less error than PT for reasonable-size databases in which more than 2 7% of the examples do not belong to the majority class Verification of the accuracy of these rules requires application to other databases and is left for future work

5 3 IDT-Based Pruning

The benefits of the intermediate decision tree (IDT) can be utilized in a cross-validation pruning method similar to that employed by Weiss and Indurkhya [1994] Weiss and Indurkhya use cross-validation to empirically determine the best cost-complexity tradeoff for pruning the decision tree The same method can be used to empirically determine the number of splits corresponding to the best IDT

Specifically, given a training and testing set, divide the training set into *k* disjoint subsets For each subset i, generate a decision tree using the other k 1 subsets and prune back the tree in a reverse breadth-first order until the error on subset *i* is minimized Record the number of splits made in this intermediate tree Repeat this procedure for each of the k subsets, and determine the average number of splits for the best IDT Then generate a new decision tree using the entire training set, but only perform the number of splits (in breadth-hrst order) as determined by the cross validation procedure

Unfortunately, initial evaluation of this IDT-based pruning strategy shows that the C4 5 pruning outperforms IDT pruning in a majority of the databases However, a closer look at the results reveals that IDT pruning consistently underestimates the number of splits needed to reach the minimum error of the intermediate decision tree Further evaluation of modifications to this IDT-based method are necessary to develop a better-performing pruner, but this is left to future work

6 Conclusions

An extensive empirical study has revealed that the best intermediate decision tree outperforms the pruned tree produced by C4 5 in a majonLy of the databases available from the UCI repository The larger number (66) of databases considered in this study allowed the detection of specific patterns in the database properties for which the intermediate decision tree will perform better than the pruned tree Therefore, a pruning strategy based on finding the best intermediate decision tree (I e , number of splits leading to this intermediate decision tree in a breadth-fir6t order) may outperform current pruning strategies in databases satisfying the detected patterns Further experimentation is required to verify this hypothesis

The relationships found between database properties and induction strategies are rare in the literature, and more work here will be of great benefit to the understanding of decision tree induction Specifically, these relationships need to be tested on artificial databases designed to verify and refine the patterns The use of the C4 5 pruning parameter optimizer provides further information from which to discern patterns between database properties and proper parameter settings

No one inductive strategy will perform well on all databases However, continued derivation and verification of relationships between database properties and induction strategies will lead to improved guidelines for selecting the proper strategy

A Databases

All databases used in the experiments were taken from the UCI Repository of Machine Learning Databases [Murphy and Aha, 1994] Table 1 lists the 66 databases and the corresponding label used for the database throughout the paper Although many more than 66 databases are available in the repository, the remainder were not included because they either were domain theories, were data generators, had continuous class values, or (in the one case of David Slate s letter recognition database) look loo long to process

Some of the dat ibases underwent minor modifications for use with C4 55 The ANN database consists of the combined examples from the data and test files, and attributes with missing values for all examples were Ignored For IMP the *Symboling* attribute was used as the class 1 if *Symboling* ≤ 0, 2 if *Symboling* > 0 For BC R, attributes *Age Tumor Size, }nv Wodes* and *Dtg Walig* were changed to continuous

Version 1 or the bridges (BR1-BR5) databases was used In each database the *Identifier* attribute was ignored values of Y for the *River* attribute were changed to [7], and fractional values of the *Location* attribute were rounded to the nearest integer The five databases (BR1-BR5) use attributes 2-8 and class attributes *l-or-d, maternal, span* re/-/ and *type,* respectively Examples with missing class values ([7] or MIL) were deleted

For HTC class values of 2 3 and 4 were changed to 1 to vleld a two-class problem The IIOR database consists of the examples from both the training and testing files Attribute 24 was used as the class attribute attributes 3, 20, 26 27 and 28 were ignored, values of 9 for the *Age* attribute were changed to 2 and values of 3 for the *Capillary Refill Time* attribute were changed to 2

The FLR dmaba&L used the flare2 data file The three class attributes describing which type of flare appeared (if any) were reduced to one binary class attribute as to whether any flare occurred In databases TII0-TH3 and TH5, attribute 28 was ignored, because all values were mssing

For the sponge (SPG) database, attribute 38 (L Papilas) was used as the class attribute for its even distribution and minimal dependence on other attributes Attribute 1 (Sponge.Name) was ignored as was attribute 37 (L Numero_De_Papilas) for its dependence on the class attribute Attributes 7-28 were typed continuous and case 65's illegal value of S1.TIPO for attribute 10 was set to 1-TIPO

The Wisconsin breast cancer database (BCW) was originally obtained from the University of Wisconsin

Table 1 Three-letter labels of databases used in the experiments

Label	Database directory/file"
ANN	an ne-ali ng / vine aJ
AUD	audiology/audiology at and archied
IMP	autoa/imports-BS
BAL	balance-scale
BA1	baJloonj/adult+st retch
BA2	balloons/adult stretch
BA3	balloons/yellow small
BA4	balloona/yellow small+adult ureidi
BCR	breast cancer
BCW	breast cancer Wisconsin
BR1	bridges/material
BR2	bndgea/re.1 1
BR3	bndges/ipan
BR4	bridge*/1 or d
BRS	bridges/type
CHS	chess/king rook vs-king pawn/kr vs-kp
CRX	credit screening/erx
FLO	flags/flag2
GLS	glass
HAR	h ayes-roth
HTC	heart diveaae/develand
HEP	hepatitu
HOR	horse-colic
SEG	image/segmentation
ION	ionosphere
IRI	ins
LAB	labor negotiations/labor neg
LEN	lenses
LDB	liver disordeni/biipo
LNG	lung-cancer
LYM	lymphography
M M	monlts-problems/monWKl
MK3	monks problems /monk- 2
MK3	monks-probkms/monks-3
MUS	mushroom/ agaric us-lepiota
PID	pima-mdians-diabetes
POP	postoperative-patient data/post operative
PRI	primary tumor
SLC	shuttle-landing control
FLR	solar flare/flare
SB5	soybean/soybean small
SBL	soybean/soybean large
ORE	space-shuttle/o-nng eroaion
LRS	peel rometer/ lra
SPG	sponge
AUS	statlog/australian
HRT	tailog/heart
SAT	tatlog/sat image/sat
SHT	tatlog/ihuttle
VEH	tatlog/vehicle
THD	thyroid diaease/allbp
TH1	thyroid disease/allhyper
TH2	thyroid disease /all hypo
TH3	thyroid diseate/allrep
TH4	thyroid diieaae/anri
TH5	thyroid disease/dis
TH6	thyroid disease /hypo thyroid
TH7	thyroid disease/new thyroid
THS	thyroid disease/sick euthyroid
TH9	thyroid disease/sick
TTT	tic tac toe
TRN	trains
SON	undocumented/connectionisl ber ch/sonar
VOT	voting records/house-voles-84
WIN	vine
ZOO	too

Hospitals in Madison thanks to Dr William H Wolberg [Mangasatian and Wolberg, 1990] The breast cancer (BCR), lymphography (LYM) and primary tumor (PRI) databases were originally obtained from the University Medical Centre, Institute of Oncology, Ljubljana Yugoslavia thanks to M Zwitter and M Soklic The Cleveland heart disease database (HTC) was originally obtained from the Cleveland Clinic Foundation thanks to Dr Robert Detrano

Table 2 lists the properties of the individual databases The number in parentheses next to the Database label is the number of different class values appearing in the data Size is the total number of available examples .Error is the classification error on all examples by guessing the most frequent class appearing in the data The next ten columns describe the non-class attributes of the database, empty entries indicate zeros in the first eight of these columns Cont is the number of continuous attributes The next seven columns show the number of discrete attributes with the indicated number of different (non-missing) values appearing in the data Miss is the number of attributes containing missing values in the data, and Tot is the total number of attributes

B Sample of Experimental Results

Experimental results for the 66 databases were generated for the optimization of the full tree, pruned tree, intermediate full tree and intermediate pruned tree The number of splits used for each tree was also recorded, but only the error was optimized The last three columns of Table 2 show a sample of the results when optimizing the pruned tree produced by C4 5 according to the hill-climbing optimization strategy described in Section 4) The first two columns show the average error and standard deviauon for 50 trials using two-thirds training and one-third testing sampled without replacement The final column shows the difference between the error of 1FT and PI opt The significance level of a one-sided, upper tailed matched pairs test is shown in parentheses, where a value of 0 05 or less indicates a significant positive difference and a value of 0 95 or more indicates a significant negative difference

References

[Elomaa, 1994] T Elomaa In defense of c4 5 Notes on learning one-level decision trees In Proceedings of the Eleventh International Conference on Machine Learning, pages 62-69, 1994

[Holder and Chaudhry, 1993] L B Holder and A Chaudhry Simple selection of utile control rules in speedup learning In Proceedings of the Third International Workshop on Knowledge Compilation and Speedup Learning, pages 77-82, 1993

[Holder, 1990] L B Holder The general utility problem in machine learning In Proceedings of the Seventh, International Conference on Machine Learningt pages 402-110, 1990

[Holder, 1992a] L B Holder Empirical analysis of the general utility problem in machine learning In Procecdings of the Tenth National Conference on Artificial Intelligence, pages 249-254, 1992

[Holder, 1992b] L B Holder Unifying empirical and explanation-based learning by modeling the utility of

Table 2 Description of database properties and sample error results when optimizing the pruned tree

Database	Size	Error	Attributes: Cont	1	2	3	4	5	6	> 6	Miss	Tot	PT_{opt}	IFT	IFT<PT_{opt}
ANN(5)	898	0.238	6	14	7	2	2			1	23	32	0.079 ± 0.022	0.042 ± 0.009	0.036 ± 0.020 (0.000)
AUD(24)	226	0.748			61	4	1	2	1		7	69	0.197 ± 0.047	0.203 ± 0.045	0.006 ± 0.013 (0.998)
IMP(2)	205	0.449	15		4	1		1		4	7	25	0.139 ± 0.047	0.111 ± 0.038	0.028 ± 0.037 (0.000)
BAL(3)	625	0.539	4								0	4	0.209 ± 0.024	0.201 ± 0.023	0.008 ± 0.014 (0.000)
BA1(2)	20	0.400			4						0	4	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000 (1.000)
BA2(2)	20	0.400			4						0	4	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000 (1.000)
BA3(2)	20	0.400			4						0	4	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000 (1.000)
BA4(2)	16	0.438			4						0	4	0.376 ± 0.205	0.308 ± 0.172	0.068 ± 0.138 (0.001)
BCR(2)	286	0.297	4		3	1		1			2	9	0.279 ± 0.035	0.256 ± 0.031	0.023 ± 0.026 (0.000)
BCW(2)	699	0.345								9	1	9	0.058 ± 0.017	0.061 ± 0.014	0.003 ± 0.011 (0.970)
BR1(3)	106	0.255	3		1	1	1			1	5	7	0.166 ± 0.061	0.128 ± 0.053	0.038 ± 0.046 (0.000)
BR2(3)	103	0.437	3		1	1	1			1	5	7	0.300 ± 0.055	0.282 ± 0.058	0.018 ± 0.038 (0.001)
BR3(3)	92	0.424	3		1	2				1	3	7	0.312 ± 0.083	0.292 ± 0.081	0.020 ± 0.076 (0.034)
BR4(2)	102	0.147	3		1	1	1			1	4	7	0.157 ± 0.051	0.126 ± 0.048	0.031 ± 0.038 (0.000)
BR5(6)	105	0.581	3		1	1	1			1	5	7	0.443 ± 0.091	0.411 ± 0.075	0.031 ± 0.042 (0.000)
CHS(3)	3197	0.478				35	1				0	36	0.006 ± 0.003	0.005 ± 0.003	0.001 ± 0.003 (0.007)
CRX(2)	690	0.445	6		4	3				2	7	15	0.152 ± 0.020	0.136 ± 0.019	0.016 ± 0.017 (0.000)
FLG(8)	194	0.691	10		12		1		1	4	0	28	0.388 ± 0.054	0.361 ± 0.051	0.028 ± 0.033 (0.000)
GLS(6)	214	0.645	9								0	9	0.336 ± 0.051	0.299 ± 0.040	0.037 ± 0.039 (0.000)
HAR(3)	132	0.614				1	3				0	4	0.278 ± 0.060	0.265 ± 0.059	0.013 ± 0.020 (0.000)
HTC(2)	303	0.459	5		3	3	2				2	13	0.269 ± 0.039	0.232 ± 0.032	0.037 ± 0.026 (0.000)
HEP(2)	155	0.206	6		13						15	19	0.195 ± 0.052	0.162 ± 0.041	0.033 ± 0.036 (0.000)
HOR(2)	368	0.370	7		3	4	5	2	1		21	22	0.146 ± 0.026	0.132 ± 0.024	0.013 ± 0.015 (0.000)
SEG(7)	2310	0.857	19								0	19	0.040 ± 0.007	0.038 ± 0.007	0.002 ± 0.003 (0.000)
ION(2)	351	0.359	34								0	34	0.095 ± 0.030	0.084 ± 0.027	0.011 ± 0.012 (0.000)
IRI(4)	151	0.669	4								0	4	0.044 ± 0.025	0.041 ± 0.025	0.004 ± 0.010 (0.009)
LAB(2)	57	0.351	8		3	5					16	16	0.182 ± 0.073	0.143 ± 0.058	0.039 ± 0.060 (0.000)
LEN(3)	24	0.375			3	1					0	4	0.155 ± 0.133	0.133 ± 0.114	0.022 ± 0.079 (0.024)
LDB(2)	345	0.420	6								0	6	0.360 ± 0.052	0.320 ± 0.036	0.040 ± 0.036 (0.000)
LNG(3)	32	0.594			13	41	2				1	56	0.524 ± 0.144	0.478 ± 0.130	0.045 ± 0.103 (0.001)
LYM(4)	148	0.453	3		9	2	3			1	0	18	0.243 ± 0.048	0.218 ± 0.050	0.026 ± 0.024 (0.000)
MK1(2)	432	0.500			2	3	1				0	6	0.010 ± 0.020	0.061 ± 0.049	0.051 ± 0.050 (1.000)
MK2(2)	432	0.329			2	3	1				0	6	0.324 ± 0.030	0.324 ± 0.030	0.000 ± 0.000 (1.000)
MK3(2)	432	0.472			2	3	1				0	6	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000 (1.000)
MUS(2)	8124	0.482		1	5	1	5	1	2	7	1	22	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000 (1.000)
PID(2)	768	0.349	8								0	8	0.262 ± 0.023	0.256 ± 0.020	0.006 ± 0.020 (0.024)
POP(4)	90	0.300	1		2	5					1	8	0.285 ± 0.060	0.277 ± 0.063	0.007 ± 0.022 (0.010)
PRI(21)	339	0.752	1		14	2					5	17	0.594 ± 0.038	0.576 ± 0.042	0.019 ± 0.028 (0.000)
SLC(2)	256	0.434			4		2				0	6	0.017 ± 0.017	0.011 ± 0.010	0.006 ± 0.012 (0.001)
FLR(2)	1066	0.169		1	4	2	1		2		0	10	0.187 ± 0.020	0.175 ± 0.018	0.012 ± 0.008 (0.000)
SBS(4)	47	0.638		14	13	3	4			1	0	35	0.026 ± 0.034	0.025 ± 0.034	0.001 ± 0.009 (0.538)
SBL(19)	683	0.865			16	12	6			1	34	35	0.088 ± 0.023	0.080 ± 0.023	0.002 ± 0.011 (0.865)
ORE(3)	23	0.304	2								0	2	0.212 ± 0.132	0.207 ± 0.135	0.005 ± 0.035 (0.161)
LRS(4)	531	0.576	100							1	0	101	0.347 ± 0.036	0.324 ± 0.029	0.024 ± 0.027 (0.000)
SPG(2)	76	0.461	2		14	9	8	3	3	4	1	43	0.172 ± 0.075	0.142 ± 0.071	0.030 ± 0.037 (0.000)
AUS(2)	690	0.445	6		4	2				2	0	14	0.155 ± 0.027	0.138 ± 0.017	0.017 ± 0.019 (0.000)
HRT(2)	270	0.444	7		3	2	1				0	13	0.257 ± 0.042	0.229 ± 0.035	0.028 ± 0.025 (0.000)
SAT(6)	6435	0.762	36								0	36	0.141 ± 0.007	0.144 ± 0.007	0.003 ± 0.005 (1.000)
SHT(7)	14500	0.208	9								0	9	0.001 ± 0.001	0.001 ± 0.000	0.000 ± 0.000 (0.000)
VEH(4)	846	0.742	18								0	18	0.293 ± 0.026	0.278 ± 0.023	0.014 ± 0.016 (0.000)
TH0(3)	3772	0.043	6	1	20			1			7	28	0.027 ± 0.005	0.026 ± 0.005	0.001 ± 0.003 (0.066)
TH1(5)	3772	0.027	6	1	20			1			7	28	0.013 ± 0.004	0.013 ± 0.004	0.000 ± 0.003 (0.909)
TH2(4)	3772	0.077	6	1	20			1			7	28	0.005 ± 0.002	0.005 ± 0.002	0.000 ± 0.001 (0.9 3)
TH3(4)	3772	0.033	6	1	20			1			7	28	0.009 ± 0.003	0.008 ± 0.003	0.001 ± 0.001 (0.001)
TH4(3)	7200	0.074	6		15						0	21	0.004 ± 0.001	0.004 ± 0.001	0.000 ± 0.001 (0.070)
TH5(2)	3772	0.015	6	1	20			1			7	28	0.013 ± 0.004	0.013 ± 0.003	0.000 ± 0.002 (0.883)
TH6(2)	3163	0.048	7		18						6	25	0.008 ± 0.002	0.008 ± 0.002	0.001 ± 0.001 (0.000)
TH7(3)	215	0.302	5								0	5	0.072 ± 0.031	0.067 ± 0.028	0.005 ± 0.011 (0.001)
TH8(2)	3163	0.093	7		18						6	25	0.023 ± 0.004	0.021 ± 0.004	0.001 ± 0.002 (0.000)
TH9(2)	3772	0.061	6	1	20			1			7	28	0.013 ± 0.004	0.012 ± 0.003	0.001 ± 0.002 (0.002)
TTT(2)	958	0.347				9					0	9	0.143 ± 0.026	0.133 ± 0.022	0.010 ± 0.017 (0.000)
TRN(2)	10	0.500	10	4	12	2	2	1		1	10	32	0.300 ± 0.280	0.287 ± 0.278	0.013 ± 0.094 (0.161)
SON(2)	208	0.466	60								0	60	0.294 ± 0.056	0.270 ± 0.048	0.024 ± 0.028 (0.000)
VOT(2)	435	0.386			16						16	16	0.037 ± 0.017	0.033 ± 0.014	0.005 ± 0.008 (0.000)
WIN(3)	178	0.601	13								0	13	0.084 ± 0.044	0.084 ± 0.043	0.001 ± 0.005 (0.161)
ZOO(7)	101	0.594	1		15						0	16	0.066 ± 0.035	0.050 ± 0.029	0.019 ± 0.026 (0.000)

learned knowledge In *Proceedings of the ML9S Workshop on Knowledge Compilation and Speedup Learn tng,* 1992

[Holte, 1993] R C Holte Very simple classification rules perform well on most commonly used datasets *Machine Learning,* 11(1) 63-90, 1993

[Mangasanan and Wolberg, 1990] 0 L Mangasarian and W H Wolberg Cancer diagnosis via linear programming *SIAM News,* 23(5) 1-18, September 1990

[Mingers, 1989] J Mingers An empirical comparison of pruning methods for decision tree induction *Machine Learning,* 4(2) 227-243, November 1989

[Murphy and Aha, 19941 P M Murphy and D W Aha UCI repository of machine learning databases Machine-readable data repository University of California, Department of Information and Computer Science, Irvine, CA, 1994

[Murphy and Pazzanm, 1994] P M Murphv and M J Pazzam Exploring the decision forest An empirical investigation of Occam's razor in decision trte in duction *Journal oj Artificial Intelligence Research* 1 257-275, 1994

[Pagallo and Haussler, 1990] G Pagallo and D Haussler Boolean feature discovery in empirical learning *Machine Learning,* 5(1) 71-100, 1990

[Qumlan, 1992] J R Quinlan *Cl, 5 Prog* rams /or *Machine Learning* Morgan Kaufmann Publishers 1992

[Schaffer, 1993] C Schaffer Overfitting avoidance as bias *Machine Learning,* 10(2) 153-178, 1993

[Schaffer, 1994] C Schaffer A conservaUon law for generalization performance In *Proceedings of the Eleventh International Conference on Machine Learn* ing, pages 259-265, 1994

[Weiss and Indurkhya, 1994] S M Weiss and N Iridurkhya Small sample decision tree pruning In *Proceedings of the Eleventh International Conference on Machine Learning,* pages 335-342, 1994