Lookahead and Pathology in Decision Tree Induction

Sreerama Murthy
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218 U S A
murthy @ es Jes edu

Steven Salzberg
Department of Computer Science
Johns Hopkins University-
Baltimore MD 21218 U S A
balzberg^c; jhu edu*

Abstract

The standard approach t decision tree in duction is a top-down greedy agonthm that makes locall} optimal irrevocable decisions at each node of a tree In this paper we empir- •call} study an alternative approach in which the algorithms use one-level loo kalie to de- ride what test to use at a node weystem- atically compare using a very large number of rfal and artificial data sets the quality of dmsion trees induced by the greedv approach to that of trees induced using lookahead The main observations from our experments are (1) the greedv approach consistently produced trees that were just as at curate as trees pro- duced with the much more expensive lookahead step and (n) we observed manv instances of *pathology,* I e, lookalnad producrd trees that were both larger and less accurate than trees produced without it

1 Introduction

Th(standard algorithm for constructing decision trees from a set of examples is greedv induction — a tree is induced top-down with locally optimal (hoices made at each node without lookahead or backup As the greed> approach can produce suboptimal trees m terms of tree siz(and depth, it is natural!} ofintest si to explore wa\s to improve the greedv strategy

Fixed depth lookahead starch is a standard technique for improving greed} algorithms [Sarkar *ct al*, 1994] Lookahead is largely unexplored in tin dmsion tree literature barring a few scattered attempts discussed in Section 5 The advantagt-S, or lack thereof of lookahead search have not been s}steinaticallv quantified in the context of decision tree or rule induction

With the rapid increases in computing power in rect nt years, limited lookahead is now feasibh for moderately large data sets The question that therefore arises is *what are the benefits (tf any) thai we might gain from employing this more costly approach*[2] In the current paper, we attempt to answer this question (impincally WE compare greedily induced trees with those induced with one-level lookahead, using two large classes of synthetic data and eight real-world data sets from the UC1 machine learning repositor> [Murphy and Aha, 1994] The results suggest that

- Limited lookahead search does not produce significantly better decision trees On average, it produces trees with approximately the same classification accuracy and size as greed} induction
- Limited lookahead s< arch produces inferior decision trees in a significant number of cases i e decision tree induction exhibits the same, *pathology* that has been observed in game trees [Nan, 1983]
- Tree post-processing techniques such is pruning arc at least as beneficial as limited lookahead for a variety of real-world data sets In this context, we describe a new post-processing technique decision tree *balancing*

Section 2 describes our experimental method SEC Lions 3 and 4 present the results of our experiments with synthetic and real world data respectivel} Section 5 summarizes related work in the literature and discusses open questions

2 Experimental method

The algorithms we used in all uur experiments, *Greedy* and *Look,* are described below *Look* performs one level of lookahead to decide what test to use at a decision Tree node while *Gretdy* decides based only on local considerations In the pseudocode below, *S* is the set of training examples where each example is assumed to compnse *i* set of nummc features and a class label

Algorithm GREED\(S)

1 If S contains examples from onl} one class halt
2 (onsider all distinct tests T of the form $r < k$ on I he features of S The *Ls* are chosen to be the midpoints between adjacent feature values C hoose the test T^* that is the best according lo a prc-d<-fin< d goodiu ss measure
3 Split S into two subsets SI and S2 using T''
4 Recursivel} run this procedure on SI and S2

Algorithm LOOK(S)

1 Execute step 1 of GREEDy
2 For each test T of the form $x < k$ do

(a) Split S into sets S] and S2 using T

(b) Find the best split of SI into sets SII and SI2 using steps 1-3 of algorithm GREEDY

(c) Repeat (b) on S2 forming set', S21 and S22

(d) (ompute the goodness of splitting S into Si 1 S12 S21, and S22 using the same goodness measure as GREEDY This is 1 s goodness

3 Execute steps 3,4 of GREEDY

We experimented with two pre-defined goodness mea Mires namely the Gini index of diversity [Breiman *e1 al* 1984] and information gain [Quinlan 1986] ' This gave us four algorithms for our experiments which we named *Greedy-Gim Greedy-Info Look Gini* and *Look-Info* Note that *Gmedy-Gini is* essentiall\ identical to tlhe C ART algorithm [Breiman *ei al* 1984] ind *Greedy Info* to the ID'i algorithm [Quinlan 198b]

Our experiments with s\nth<tic data (Section 3) systematically compare the trees induced with one level lookahead to those induced greedily, *our entire clasies of decision trees,* ~ We define below two classes of decision trees that are small enough to be amenable to systematic experimentation on the entire clss and general enough to be interesting We first generated a training set *TRAIN* and a test set *TEST 1 R A] * has 500 exam pies and TEST 5000 examples with two real valued attributes for each example All attribute values were generated uniformly al random in the interval (0 10) The same unlabeled training and test sets art used in all the experiments Each experiment tested a different element of the concept class and the eximples were albeled accordingly Trees built on *TRAIN* were tested on IEST for every concept in each class

Trees are compared to each other throughout this paper using three quality measures — accuracy size and depth Accuracy is the percentage of correct classification on *TEbT* Size is the. number of leaf nodes Depth is the length of the longest path in the tree

3 Experiments with Synthetic Data

3 1 *C* Exhaustive vs greedy search

We designed our first set of experiments to measure how close lo optimal are the trees produced by greedy induction on a fixed concept class More precisely, we consider a class of concepts *C* in which one-level lookahead is equivalent to exhaustive search, for this class lookahead always gives us the optimal tree while greedy induction may not We systematically evaluate the effectiveness of greedv induction over this entire class

C is a class of binary decision trees defined as in Fig 1 and has a total of 5844 distinct trees (Frees that are equivalent except for having their class labels swapped are not considered distinct) One level of lookahead from the root in class (. will always hnd the ophmal decision tree in terms of both size and depth Frees in this class realistically occur in many situations as subtrees of a larger tree and it is reasonable to ask if we should con stantl} check one level ahead while building such a tree in order to see if we can finish off a subtree Because e\en one level of lookalit ad is ver> costl}, we wish to quantily lis possible advantages More specifically the complexit} of the standard greedy algorithm is *0(dn* logn) at a node, for *d* allnbutes and n examples One level lookalit ad has complexity $O(d^2n^2)$

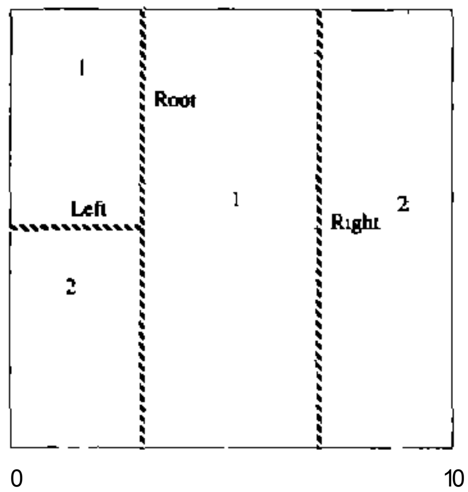


Figure 1 Class *C* consists of all balanced decision tries on a 10 X 10 grid such that each tree has three lepit (internal) nodes and all test nodes are non-Invial, in the sense thai they split heterogeneous point sets IS There are two classes 1 and 2

Using the experimental method defined in Section *I* we built 5844 trees on the set TRAIN with each of lh< four algorithms I hus one tree was, induced by each algo nlhm for every possible element of *C* As ont level lookd head is the same as exhaustive search on *C* *Look-Info* and *Look-Gim* produce identical trees Figure *2* summarizes the differences between the decision tries induced by *Grftdy-Gim Greedy Info,* and exhauslive search (u ther *Look-Info* or *Look* Gtni) over the entire class *C The figure* figure shows the mean and one quartile ranges of the accuracy, tree size and maximum depth (One quartih range is the interval that includes 25% of the samples above and below the mean)

As the figure shows the differences between *Giecdy-Gtm Greedy Info,* and *Look* are quite small, in spite of the fact that greedy induction uses only about () 004 limes as much search as exhaustive search The average number of candidate splits evaluated per tree in *C* art. *Grtedy-Gim* 1798, *Greedy-Info* 1718, *Look* 419,301 The differences in accuracy between the greedy algorithms and *Look* are negligible The difference in tree size between *Greedy-Info* and exhaustive search is 0 36 nodes, less than one standard deviation The difference of 0 63 between the average tree size of *Grcedy-Gini* and *Look* is slightly more pronounced but still not significant The only measure for which greedily induced trees are significantly worse than the optimal trees is maximum depth Exhaustive search produces trees whose longest

We chose Gini index and information gain because They have been widely used for real world applications Expen mente with other goodness measures may be interesting, but we suspect the results would be similar

[2]Thie style of empirical investigation is made possible by the existence of extremely fast inexpensive computers See [Murphy and Pazzani 1994] for another example of this sLyle

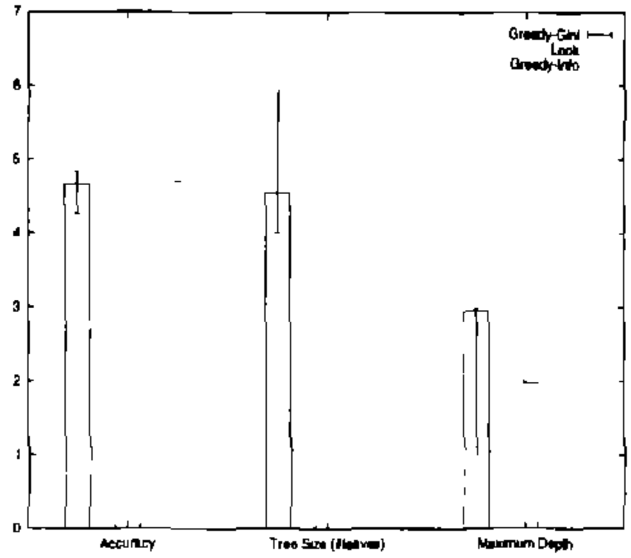


Figure 2 Summary of experments Willi class *C* *T*he mean and one quartilt ranges for accuncv tree size and maximum depth are shown for *dntdy Gam* one-level lookahead and *Greedy-htfo* The accuricies shown are the amounts above a baseline value of 95%

piths an on average one level] shorter thin what is product d with the greed} algorithms '

Figure { shows the effects of one level lookahead (equivalent!} exhaustive search) for class *C* in more de tail The horizontal axis plots the impiovement due to lookahead The(line for ACCURACY shows the ;INcrease in arcurac} whereas the lines for tres size and depth show the *decrease* in these measures when lookahead is used The ve rtical axis plols the number of trees in which lookahead causes a particular improvement hur points on \ — 0 lookahead had no effect and for points to the right of \ = 0 lookahead was beneficnl Tor points to the left of (he line 1i = It greed} induction was *better* thin lookahead Only tin measurement for information gain are shown due to space constraints

I ig i offers several interesting insights First each of the three lines has a single prominent peak The peaks it \ — 0 for accuracy and tree size lines show that for a large number of trees lookahead did not make *any* difference in terms of these measures The depth peak at \ = 1 shows that the maximum d<pth of *most* of the greedily induced trees is exactly OIK more than optimal lo understand why the greedy approarh builds trees with unnecessarily long paths we looked at several of these trees individually and found that man} trees were unbalanced That is, there were several trees in which nodes could bi moved around without altering the original partitioning and accuracy to cut short the maximum depth of the tree Appendix A describes a simple post-proceasing step to rebalance a greedily m-

3Note that the effect of lookahead on average or expected depth may not be the same as that on maximum depth The expected depth of a greedily induced decision tree has been observed to be very close to that of the optimal tree [Murthy and Salzberg, 1995]

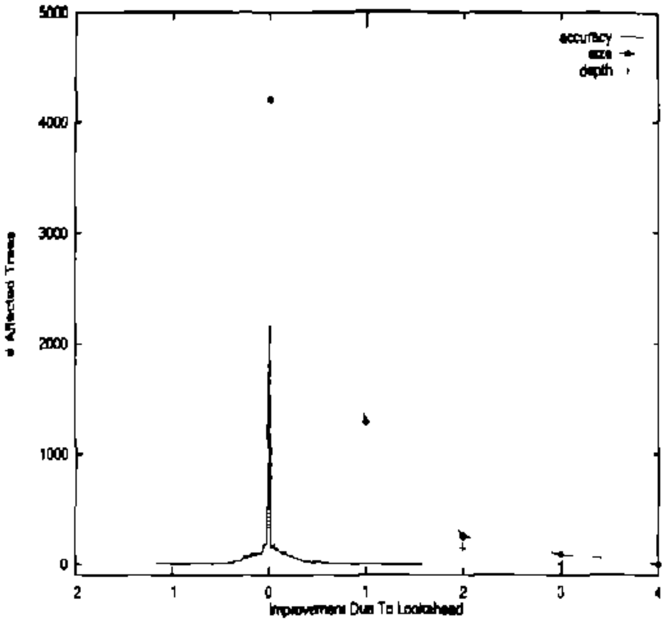


Figure \ Elfect of one level lookahead in trees produced with information gam for class *C* Improvements in accuracy, si^e and maximum depth are shown, along with the number of trees in which these improvements occur Negative values on the X-axis mean that iookahead produced inferior trees

duced tree, in order to reduce its worst-case classification cost Use of this decision tree *balancing* procedure filled some of the gap between the greedy and lookahead trees in all our experiments

Second, it is interesting to note that lookahead ac lually hurts accurat\ in almost as many trees as those in which it enhances accuracy This property where lookahead search finds inferior solutions is known as *pathology* in the context of game trees [Nau 1981 Mutchler 1993] We discuss pathology for decision trees further in Section 3 2 where this t rend is exhibited more prominently Pathology cannot occur for tree size or depth for class c1, because one-level lookahead is equivalent to exhaustive search However our next class *Cs* includes deeper trees and limited lookahead can and does produce trees that are worse in terms of size and depth

Third, we tan see from Figure 3 that there are some greedily induced trees that have as manv as 4 leaves more than the optimal We looked at all such large trees, and found that thev always had several "mimmallv useful splits, splits that were separating very few points Such splits cin be easily avoided with a simple stop-splitting rule, narrowing the gap between lookahead and greed) induction further

3 2 C$ A class of larger trees

Thi6 section extends class *C* to a class *Cs,* which contains slightly larger trees Each tree in *Cs* is obtained from a different tree in C, as follows

1 Remove *T* from *C*

2 Randomly choose a leaf node *L* of *T*

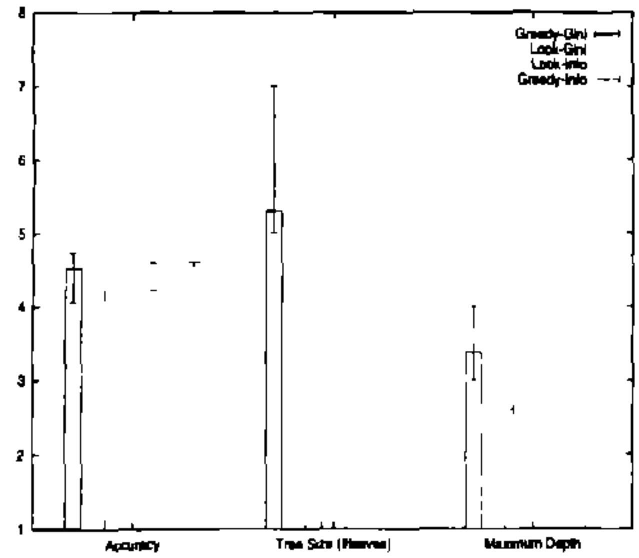


Figure 4 Summary of experiment with class *Cs* The mean and one quartile ranges for accuracv tree size and maximum depth are shown for *Greedy (Gins, Look-Gim Greedy-Info* and *Look-Info* The accuracies shown are the amounts above a baseline value of $^{()r}$->%

J Split *L* with a randomly chosen non-lmial spill *S of the form *x,* < *k* where it is an integer in the range (0,10) If no valid split exists, go to step *l* and choose a different *L*

4 Assign one side of *$* to class 1 randomlv and assign the other side to class 2

5 Add *T* to *Cs*

Each decision tree in *Cs* is a binary trt-< with four test (internal) nodes and has a maximum depth of 1 For these trees one level lookahead is not sufficient to find the optimal tree Note that while *Cs* has 5844 trees the same as C, another run of the above procedure would create a different definition of cs because of the ran domized steps Using exhaustive enumeration in place of these random choices would produci a class that is vastly larger too large for systematic t expenmentation The experimental method used for *Cs* was identical to that used for *C* One important difference is, since one-level lookahead is not equivalent to exhaustive search on *Cs, Look-Gmi* and *Look Info* do not produce identical trees for this class

The experimental results with class *Cs* strengthen the conclusions drawn from experiments with class *C* Figure 4 summarizes the differences in accuracy tree size and maximum depth between *Greedy-Gim, Look-Gtni, Greedy-Info* and *Look Info* on class *Cs* It can be seen that there is no significant improvement in accuracy due to lookahead The differences in accuracy due to lookahead are actually smaller here than they were for class C, despite the fact that the relative cost of lookahead search was higher for this class The average number of candidate splits considered per tree in *Cs* were *Greedy-Gmi* 1952, *Greedy Info* 1847, *Look Gini* 745,689 and *Look-Info* 747,037 Despite these enormous differences in computational effort, the differences in tree size are

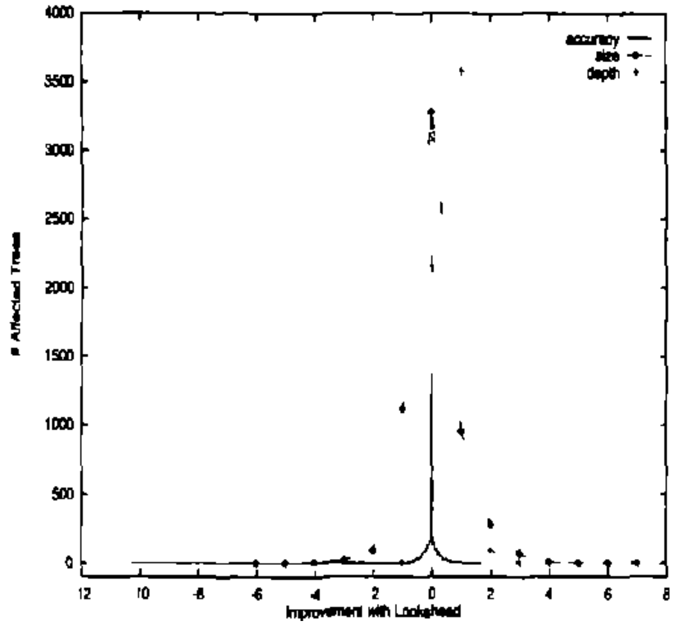


Figure 5 Effect of one level lookahead for trees in class *Cs* Improvermnte in accuracy size and maximum depth of trees builL using *Look-Info* versus *Greedy-Info* art shown Negative values on the x-axis mean that lookahead produced inferior trees

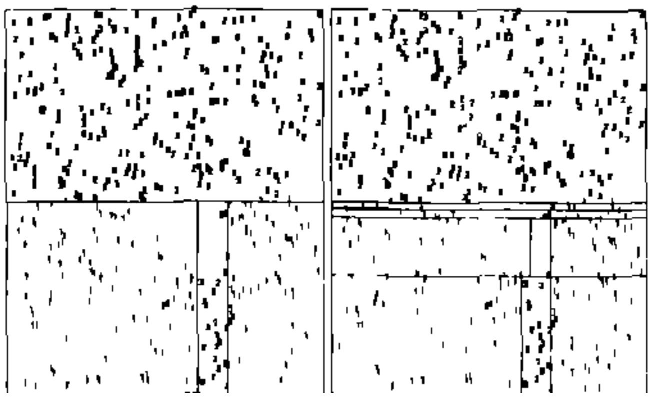
less than one standard deviation The onlv quantity for which one-level lookahead caused any noticeable improvement was maximum depth where trees were on average 0 C levels shallower when lookahead was used

Pathology results Figure 5 shows the effect of one-level lookahead for class *Cs* m more detail for *Greedy-Info* The svnlax of this figure is the same as that of Figure 3 i e points to the left of 0 on the horizontal axis represent instances of pathology, where lookahead was worse than no lookahead Lookahead hurt accuracv for a large number of trees in *Cs* just as it did for *C* In addition it produced worse trees in Lerms of tree size and depth Figure 6 shows a data set in *Cs* for which information gain exhibits pathology in terms of accuracy, size *and* depth

4 Experiments with Real World Data

In addition to the synthetic data, we also experimented with eight real world data sets, for which the underlying concepts are unknown We augmented our algorithms (*Greedy-Gini, Look-Gtni, Greedy-Info* and *Look-Info)* with pruning for these experiments, using cost complexity pruning with the one standard error rule [Breiman *el al* , 1984], reserving 10% of the training data as the pruning set All results for real world data are averages of ten 5-fold cross validation experiments

The choice of the domains is important If a greedy method can induce a highly accurate, concise classifier for a domain (e g , the well-known Ins data), lookahead is not likely to produce significant benefits We used a survey of results [Holte, 1993] to choose six "difficult" domains for our experiments - domains for which the best



Figurre 6 A pathological tree For Lhe tree on The left, induced without lookahead accuracy = 99l 74%, size — 4 maximum depth = i and lhe number of tests considered was 1545 For the free on the right, which was huilt using lookahead accuracy = 99 1*0% size* = 10, maximum depth = 4 and the number of eandidate t<sts was 1 155 901

known accuracy is aL mosl 90% 1 houeji (he low accuracies may be due to factors other than the inadequacy of greedy induction such as an overly small or noisy training set there is no sir ughtforward w *y of knowing l his *a piwn T*he six difficult domains an the breast canc(r ncurrence datahase (Bi( ) the (leveland heart disase data (C L) [UCI cleyeland dati] glass identific ation data (CTL), hepatitis diagnosis (JIT) C inadian la hor negotiations data (LA) and lymphography diagnosis (L^) [0(l-lymph-dati] In addition In these domains we experimented with l un\an in ls (\0 and \1) of the congressional voting records, clatd vpis is used In Norton [Norton 1989] for his lookahead <\penments The \] data [Holle 1903] is identical to (the \0 data e\-(<|)t that the best attribute physici in ft < freeze is removed All the data se|s were taken from the t (l Machme Learning repository [Murph\ and Aha l994] Our abbrevations for the dattstts an consistent with those of Holle [Holle 1993]

All experimental resuits reporttd in this section were obtained with information gam Resuits with Gmmdex look very similar and art omitted for space considerations Figures 7 and 8 summarize Lhe results for accuracy and tree size respective]} l he plot for tree depth looks almost identical to that for tree size and is omitted In each figure we plot the values of lhe measure obtained using four induction methods (1) *Greedy-Info,* (n) *Look-hifo* (in) *Greedy-Info* with pruning and (iv) *look-Info* with pruning There are eight linos in each figure corresponding to the eight data sets

Consider the accuracy plot m Fig 7 The first obser vation is that the accuracies do not vary much between various induction methods On closer observation, accuracy drops for six out of the fight databases (all except VI and Gt) when lookahead is used In addition *Greedy-Info* with pruning produces more accurate trees than *Look Info* for five data sets Pruning almost always (7 out of 8 times) works better when it is used without lookahead, as can be seen from the third and fourth columns Our overall conclusion from tin accuracy plot

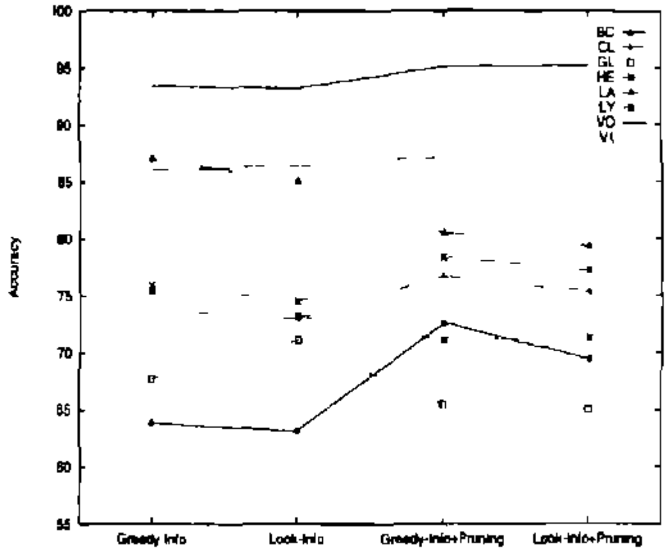


Figure 7 Lffect of one level lookahead on classification accuracy for eight real-world databases Tht accuracies with and without lookahead and with and without pruning are shown for information gam

in Fig 7 is that lookahead doesn t affect accuracy signifi cantly for these domains and that pruning is both much cheaper and more effective at creating accurate trees

Now consider the tree size plot, shown in Fig 8 Lookahead does reduce the tree size, bv a small amount in most domains These benefits however, are overshadowed by. the benefits of pruning For all domains except the I A data (which has a verv small tree to begin with}, pruning helps produce substantially smaller lrees than lookahead

The results of our experiments with real data support our results with the artificial data [4] Limited lookahead did not help significantly in terms of classification accu rac>, size or depth, despite the fact that it is enormously mon expensive It helped produce shallower trees, but tree post-processing techniques much less expensive than lookahead (pruning in this case) were, adequate to reap comparable if not larger, benefits Finalh both of the goodness measures we used (Gini index and information gain) exhibited pathology on the real world domains also

5 Discussion

Several versions of the optimal decision tree induction problem are known to be NP-Complete [Hvafil and Pivest, 1976 Murph) and McCraw 1991] As a re suit, virtually all implemented decision tree systems use a heuristic greedy approach There have been however some exceptions to this rule Mont [Morel 1982] surveys early induction systems that used dynamic

[4] Note that ill of our "difficult" data sets happen to he quite small, probably inherently inadequate for learning The experiments with real data are given only to substantiate the earlier observations on the artificial data We would not make strong conclusions from the UCI data alone

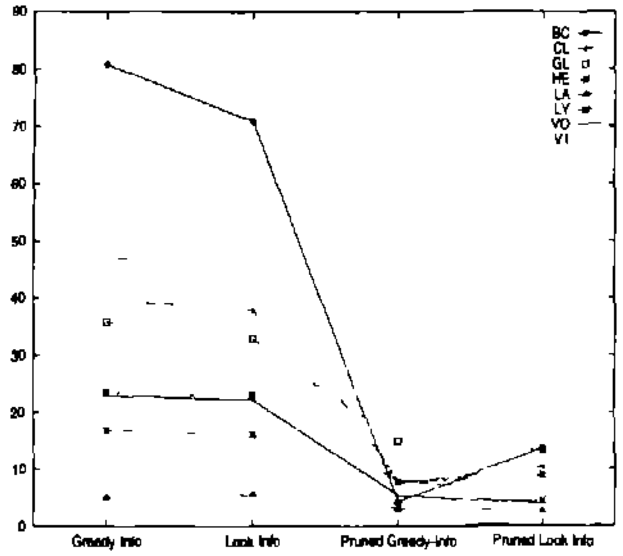


Figure 8 Effect of one level lookahead on tree size for eight real world databases The tree sizes with and with out lookahead and with and without pruning are shown for information gam

programming and brancli-and-bound methods to prodme optimal trees Hart mann at al [Hartmann of *al* 1982] describe Generalized Optimum Testing Algorithm (GOTA) an algorithm based on an information the retic criterion between branching levels in a tree With the appropriate parameter settings GO I A can do fixed-depth lookahead, different depths of lookahead al differ ent branching levels or even exhaustive search Though Hartmann et al did offer a concist framework for doing arbitrary level lookahead the\ did not evaluate the of-fects fects of lookahead on tree quality The ideas in GOTA motivated Norton & IDX system [Norton 1989] which is a variant of Qumlan & ID3 that performs lookahead Norton conducted experiments on the congressional voting records database (see Section 4), and found that lookahead reduced decision tree depth on average With a few exceptions though the advantages of lookahead were very small in Norton s experiments Ragavan and Rendell considered using lookahead for feature construction in symbolic domains [Ragavan and Ren dell 1993], and pointed out that lookahead is beneficial when there is concealed attribute interaction

The emphasis of the current paper differs significantly from the existing work on lookahead hirst our experiments are aimed to offer insights on whether or not to use lookahead when little is known about domain characteristics or attribute interactions Second, though existing papers do contain some remarks about whether lookahead did or did not help, no work has vet attempted to systematically quantify how often lookahead helped how often it did not make a difference and how often it *hurl* tree quality

The pathology results are particularly interesting, since they have not been previously reported for decision trees Intuitively, doing more search (lookahead) should produce better decision trees, just as deeper search in game trees (e g for chess) produces better game-playing programs However it has been observed that for some games, deeper search can actually produce an inferior program both with two players [Nau, 1983] and with multiple players [Mutchler, 1993] Decision trees, one can argue, are analogous to a one-player game tree Our discovery that deeper search can lead to inferior decision tree[5]; thus extends the earlier pathology results to a new domain

It is possible that pathology is a side-effect of the way heuristic goodness measures are defined Greedy methods grow a decision tree by optimizing entropy or class-divergence based measures at each node of the tret Our pathology results indicate that each such optimization i> not necessanl) improving the tree globally in terms of generalisation accuracy, tree size or depth Goodman and Smyth [1988] showed that greedily maximizing the average mutual information should result in trees thai are near optimal in terms of average depth Although our experimental results are consistent with this work pathologically deep trees indicate that locallv optimizing information gain can in fact make a tree deeper

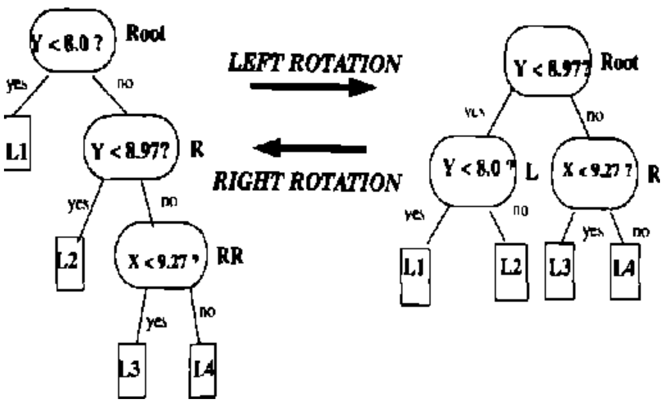
We considered only one-level lookahead in this paper One can attempt to evaluate the benefits of lookahead as a function of se arch depth We feel that such a syslematic evaluation is not only going to be computationally prohibitive, butl also probably not very useful Norlon [Norton 1989] presents e-xpenments comparing one and two level lookahead on one data set

Observing incidences of pathology (as we did in this paper) is only The first st< p in several interesting research directions Concept i lasses for which a particular good ness measure exhibits pathology can be sludied, analytically or quantitatively to determine *when* pathologv might occur On the other hand, one can attempt to isolate characteristics of data which have bearing on when lookahead is likely to help As we have only studie-d two concept classes, several othe r interesting concepts remain to be explored Another interesting question for further study is whethtr there exist effective goodness measures that guarantee no pathology

A Decision Tree Balancing

I he main benefit of lookahead search for classes *C* and *Cs* was that lookahead produced trees with shorter longest paths On closer observation, we found that several greedily induced trees had identical partitions as the ones induced with lookahead, but the latter were shallower because the trees were better balanced This trend suggests the following problem Given a decision tree *D* for a training set *TRAIN* we want to produce a tree *DR* that induces the same partitioning as *D* on *TRAIN,* but has less worst-case cost (or maximum depth)

Although little work has been done on balancing decision trees, a great deal of research has considered balanced search trees (e g [Nakamura *ct al* , 1993]) Roughly speaking, this literature deals with techniques to restructure search trees when elements are inserted or deleted, in order to restrict the depth of these trees to a logarithmic function of the number of search keys An axis-parallel decision tree in a continuous space can be



Figu e 9 Left and right rotations of a binary decision tree dotation operators can help reduce the expected rosi of classification of a derision tree without changing lis acuracy The leaf nodes I 1 L2 *tt*< in this figure can IK replaced with arbi(rar\ subtrees

interpreted as a inulti dimensional binay] search tree Such in interpretation makes il possible to use search tree balancing techniques on decision trees

The mam primitives used for rebalancing a trte in balanerd seach tree methods are *rotations* Rotations are operations in which the parent child links of some nodes in the tree art rearranged locally while guaranteeing that the functionality of th< whole from romans invarant anl *t have* adopted two simple tree rotation operators left rotate and right rolate to decision trees Thest operators are illustrated in Tigure 9 \\r found that a heuristic top-down tree balancing procedure using rola lion operators recursively at the tree nodes significantly reduces the maximum depth of greedily induced trees for classes £ and *C&*

References

[Breiman *el al* 1984] L Bremian J Fredman ()1 shen and C Stone *Classification and Regression Frees* Wadsworth Intl Group 1984

[Ilartmann *ct al,* 1982] (R P Ilartniann PK Varshney KG Melirotra, and CL Gerbench Application of information theory to the. construction of < fhcient decision trus *IEEE Trans on Info Theory* IT-28(4) 565 577 1982

[ilolte, 1993] R Holtt \ery simple chssification rules perform well on most commonh us(d datasots *Machine Learning,* 11(1)63-90, 1993

[Hyah] and Rivest 1976] L Hyafil and RL Rivest Constructing optimal binary decision trees is NP-complete *Info Proc Letters,* 5(1) 1 5-17 1976

[Moret, 1982] B M E Morel Decision trees and diagrams *Computing surreys* 14(4) 593-623, 1982

[Murphy and Aha, 1994] PM Murphy and D Aha UCI repository of machine learning databases FTP from les uci edu in the directory pub/machine learning-databases 1994

[Murphy and McCraw, 1991] O J Murphy and R L McCraw Designing storage efficient decision trees *IEEE Trans on Comp* , 40(3) 315- (19, March 1991

[Murphy and Pazzani, 1994] PM Murphy and M J Pazzani Exploring the decision forest An empirical investigation of Occam s Razor in decision tree induction *J of Art Intel Res* 1257-275 1994

[Murthy and Salzberg, 1995] S K Murthv and S Salzberg Decision tree induction How effectivt is the greedy heuristic[7] In *1st Inil Conf on Rnowledgt Discovery in Databases,* Montreal, Canada, 1995

[Mutchler 1991] D Mutchler 'The multi-player version of minimax display game pathology *Artificial Intelligence, 6*4(2) 323-336,1993

[Nahmura et *al* , 1993] Y Nakamura S Abe Y Ohsawa and M Sakauchi A balanced hierarchical data ▪structure for multidimensional data with highly efficient dynamic characteristics IF*EE Trans on knovl edge and Data Engg* 5(4) 682-691 1993

[Nau, 1983J D S Nau Decision quality as a function of search depth on game trees *J of the ACM,* J0(4) 687-708, 1983

[Norton 1989] S W Norton Generating better decision trees In *11th Intl Joint Conf on Art Intel* pages 800 805 1989

[Qmnlan 1986] J R Qmnlan Ind uction of decision trees *Machine Learning* 1 81-106, 1986

[Ragavan and Rendell, 1993] H Ragavan and I Rendell Lookahead feature construction for learning hard concepts In *Machine Learning Proceedings of the 10th Inil Conf* pages 252-259 1993

[Sarhar *ct al* 1994] U K Sarkar PP Chakrabarti, S Ghose, and S (DeSarkar Improving greedy algorithms b> lookahead-search *J of Algorithms* 16(1)1-23 1994

[UCI-cleveland-data] Cleveland heart disease data Collected bv Roberto Detrano V A Medical centtr, I ong Beach and Cleveland C lime Foundation

[HCI-lvmph-data] Lymphography data Obtained from the Uniy Med Centre Institute of Oncology, Ljubljana Yugoslavia Data provided by M Zwitter and M Soklic