Information Gain Versus Gain Ratio: A Study of Split Method Biases

Earl Harris Jr.
The MITRE Corporation/Washington C^3 Center
7515 Colshire Dr. McLean, Virginia 22102-7508[*]
esharris@mitre.org

Computer Science Department William & Mary
Williamsburg, VA 20171
esharr@cs.wm.edu

Abstract

One corollary of the Cullen Schaffer's Conservation Law of Generalization Performance indicates that no learner is generally better than another learner. If the first learner performs better than the second learner on some learning situations, the first learner must perform worse than the second learner on other learning situations. Unfortunately, the corollary does not provide a description of the circumstances where a specific learner has an advantage.

This article focuses on two decision tree learners. One uses the information gain split method and the other uses gain ratio.

It presents a predictive method that helps to characterize problems where information gain performs better than gain ratio (and vice versa). To support the practical relevance of this research, it shows that the predictive method works effectively on the contraceptive method choice problem from the Cal-Irvine Machine Learning Repository.

This article brings new insight on how these two split methods affect a decision tree learner's bias.

1 Introduction

The decision tree learner is a popular machine learning algorithm. While the technique is relatively simple to understand, researchers have successfully applied decision tree learners to such real world problems as medical insurance fraud.

A vital component of a decision tree learner is its split method. An effective split method helps the learning algorithm build a hypothesis with high pre-

dictive accuracy. Quinlan's ID3, an early decision tree learner, initially used the information gain split method. But Quinlan discovered that information gain showed unfair favoritism toward attributes with many outcomes. Consequently, gain ratio later became the default split method. [4, page 23]

Though c4.5, a descendent of ID3, still defaults to the gain ratio criterion, the user has the option to request information gain. [4, page 86] Unfortunately, it is unclear how switching to information gain affects the learner's bias.

One corollary of Cullen Schaffer's Conservation Law of Generalization Performance gives researchers a better understanding of an issue concerning the comparison of learners under certain situations. A *concept learning situation* is a triple

$$cls = (D, P, V),$$

where

D is an input space (a non-empty, finite set),

$P : D \rightarrow [0.0, 1.0]$ is an input space probability distribution, and

$V : 0 \ldots |D| - 1 \rightarrow [0.0, 1.0]$ is a vector of probabilities.

A concept learning situation defines a discrete ran-dom experiment whose sample space is the set of cases $(D \times \{0, 1\})$. A *case* consists of a domain element and a label. Furthermore,

$$(\forall d_i \in D) \Pr(\{(d_i, 1)\}) = P(d_i) * V(i).$$

The ith component of the *positive label probability vector*, $V(i)$, is the likelihood of assigning a positive label to the ith domain element.

The following corollary uses generalization accuracy as its metric for evaluating learners. Recall that the training sample of a concept learner is a case list. This metric focuses on the domain elements that are *unseen* with respect to some training sample C (i.e., they are not the domain element of any case in C). Informally, the *generalization accuracy of learner $\mathcal{L}$ under concept learning situation (D, P, V) and training sample C* is the likelihood that the hypothesis $\mathcal{L}(C)$ correctly classifies an arbitrary case whose domain element is unseen. The *generalization accuracy of learner $\mathcal{L}$ under concept learning situation (D, P, V) and training sample size $n \in \mathcal{N}$,*

$$\mathsf{ga}_{((D,P,V),n)}(\mathcal{L}),$$

is the weighted average of the preceding metric, as we vary the training samples of length n. It measures the ability of a learner's hypothesis to correctly

classify unseen domain elements, when given an arbitrary training sample of length n.

The analysis of learning situations often involves a special subset of the domain elements. The *occurring domain element set with respect to D and P is*

$$\mathsf{occurring}(D,P) = \{d \in D \mid P(d) \neq 0.0\}.$$

These are the elements that may occur within an arbitrary trial of the case experiment.

Corollary 1.1 (No Strict Improvement) *Let us consider the following.*

1. *An arbitrary finite, non-empty set D.*

2. *An arbitrary probability distribution*

$$P : D \to [0,1].$$

3. *An arbitrary training sample size $n \in \mathcal{N}$.*

4. *Arbitrary learners $\mathcal{L}_0$ and $\mathcal{L}_1$.*

If $n < |\mathsf{occurring}(DOM, pd)|$, then

$$\int_V \mathsf{lga}_{((D,P,V),n)}(\mathcal{L}_0) - \mathsf{lga}_{((D,P,V),n)}(\mathcal{L}_1) \, dV = 0.0.$$

Simply stated, one learner is not strictly better than another (with respect to generalization accuracy). [6, page 262] The learner that beats another learner in some situations must lose to that learner in other situations.

It is unfortunate the no strict improvement corollary does not provide a description of the circumstances where a specific learner has an advantage. Since there is no universally superior concept learner,

there is a need to compare the strengths and weaknesses of concept learners. The next section provides a framework for making comparisons.

2 Predictive Methods

Given an arbitrary situation experiment S, arbitrary $n \in \mathcal{N}$, and two arbitrary learners $\mathcal{L}_1$ and $\mathcal{L}_2$, a *predictive method* is a function where

$$\beta(S, n, \mathcal{L}_0, \mathcal{L}_1) \quad \in \quad \{\mathsf{L0\_wins}, \mathsf{L1\_wins}, \mathsf{tie}\}$$

A predictive method partitions the cross product of concept learning situations and training sample sizes into three sets. Each set represents environments where one learner has allegedly higher generalization accuracy than the other. This article presents a predictive method for the given decision tree learners that helps us compare their bias.

3 Information Gain

It is often advantageous to represent the domain of a learning problem as a collection of fixed-length tuples. A *test* is a mapping of domain tuples to finite set of values, called *outcomes*. Usually, a test uses the value of a single *attribute* (tuple component) to determine an outcome.

The following example comes from the Cal-Irvine

0. *Wife's age:* $\{16, 17, \ldots, 49\}$.

1. *Wife's education:* $\{\mathsf{low}, \mathsf{med-}, \mathsf{med+}, \mathsf{high}\}$.

2. *Husband's education:* $\{\mathsf{low}, \mathsf{med-}, \mathsf{med+}, \mathsf{high}\}$.

3. *Number of children ever born:* $\{0, 1, \ldots, 16\}$.

4. *Wife's religion:* $\{\mathsf{non-islam}, \mathsf{islam}\}$.

5. *Wife working status:* $\{\mathsf{employed}, \mathsf{unemployed}\}$.

6. *Husband's occupation:* $\{\mathsf{low}, \mathsf{med-}, \mathsf{med+}, \mathsf{high}\}$.

7. *Standard of living:* $\{\mathsf{low}, \mathsf{med-}, \mathsf{med+}, \mathsf{high}\}$.

8. *Media exposure:* $\{\mathsf{adequate}, \mathsf{inadequate}\}$.

Table 1: *The domain of the contraceptive method choice problem.*

attribute	information gain
0	0.045
1	0.044
2	0.018
3	0.113
4	0.004
5	0.001
6	0.006
7	0.018
8	0.015

Table 2: *Information gain scores for the contraceptive method choice data set (Does the woman use contraception?)*

Machine Learning Repository. [1] This data set, which Tjen-Sien Lim created, is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey.

Example 3.1 *For the contraceptive method choice problem, the attributes are presented in table 1.*

The data set has 1473 cases and no case has a missing value.

Table 2 presents each attribute and its score, when the training sample is the entire data set. It follows that the information gain split methods selects attribute 3, "Number of children ever born", to be the root node.

Consider trying to use a sample of the data set to

attribute	information gain
0	0.771
1	0.495
2	0.281
3	0.571
4	0.079
5	0.020
6	0.020
7	0.210
8	0.144

Table 4: *Information gain scores for a contraceptive method choice sample (Does the woman use contraception?)*

learn how to correctly classify an arbitrary element of the data set. Table 3 (on page 5) is a random sample consisting of 10 arbitrary picks from the 1473 cases. Table 4 presents each attribute and its score for the small sample. Here, the information gain split method selects attribute 0, "Wife's age", a different attribute with many more outcomes.

In this example, the sample scores for attribute 0 is inflated. Low label entropy in a partition element should imply the functional dependency. However, a test with many outcomes often partitions a training sample so finely that low entropy may be achieved only because the partition elements are small. In example 3.1, attribute 0's sample partition has 1 element of cardinality 2, 8 elements of cardinality 1, and 25 empty elements. Building decision trees using tests with inflated scores can hurt the hypothesis' generalization accuracy.

4

0	1	2	3	4	5	6	7	8	
28	high	high	1	islamic	unemployed	medium+	medium-	adequate	t
47	high	high	5	islamic	unemployed	low	med+	adequate	t
16	med-	high	1	islamic	unemployed	med+	high	adequate	f
48	high	med+	7	islamic	employed	med-	high	adequate	t
31	med+	high	2	islamic	unemployed	low	med-	inadequate	f
22	low	med-	2	islamic	unemployed	med+	high	adequate	t
47	low	low	6	islamic	employed	med+	low	adequate	f
36	med+	high	4	non-islamic	employed	med-	high	adequate	t
31	med+	high	0	islamic	employed	med-	med+	adequate	f
41	high	high	8	islamic	unemployed	low	high	adequate	t

Table 3: A Sample of contraceptive method choice data set (Does the woman use contraception?)

4 Gain Ratio

To correct this favoritism, gain ratio does the following. For the tests with above average gain scores, divide the test scores by the test's *split information*, the test outcome entropy. Then, pick from among the tests with the maximal gain ratio score. This split method doesn't select a test with an equal or below average information gain and a low split info, because a low split info doesn't provide any intuitive insight into class prediction.

Regarding example 3.1, the application of the gain ratio split method to table 3 selects attribute 1, "Wife's education", an attribute with fewer outcomes than "Wife's age" (see table 5 on page 6).

Besides correcting the unjustified favoritism of information gain, gain ratio also has a lesser known effect; it favors the creation of an unbalanced tree. In an unbalanced tree, there is large variance between

T_0	0	1	cardinality
v_0	0	10	10
v_1	30	10	40
sum	30	20	50
T_1	0	1	cardinality
v_0	0	10	10
v_1	3	1	4
v_2	3	1	4
v_3	3	1	4
v_4	3	1	4
v_5	3	1	4
v_6	3	1	4
v_7	3	1	4
v_8	3	1	4
v_9	3	1	4
v_{10}	3	1	4
sum	30	20	50

Table 6: The contingency tables with equal information gain and varying split information

the depths of the leaves.

Suppose there is a concept learning problem where applying tests T_0 and T_1 to the training sample builds the contingency tables in table 6 respectively. Test T_0 partitions the training sample into a big element and small element. A tree with T_0 in the root node will build one leaf at depth one. And since the

attribute	information gain	split info	gain ratio
0	0.771	2.922	0.264
1	0.495	1.846	0.268
2	0.281	n/a	n/a
3	0.571	2.922	0.195
4	0.079	n/a	n/a
5	0.020	n/a	n/a
6	0.020	n/a	n/a
7	0.210	n/a	n/a
8	0.144	n/a	n/a

Table 5: Information gain and gain ratio scores for contraceptive method choice sample (Does the woman use contraception?)

cardinality of the impure partition element is large, the decision tree learner should build a tree whose remaining leaves are at a much lower depth. Test T_1 partitions the training sample into many small elements. A tree with T_1 in the root node will build one leaf at depth one. However, since the impure partition elements are small, the decision tree learner should build a tree whose remaining leaves are at a shallower depth.

Both contingency tables yield the same information gain score (0.322). It follows that the information gain split method shows no favoritism to either test. However, since the split information for T_0 (0.722) is smaller than the split information for T_1 (3.379), the gain ratio split method favors T_0 (0.446 > 0.095). It follows that, when compared to a decision tree learner that uses information gain, that same learner using gain ratio is more likely to build an unbalanced tree.

How does this affect the decision tree learner's bias? A decision tree learner can be viewed as a hill-climbing search algorithm, where the split method looks for local elevations in the search topography. The search usually stops when the sample becomes too class pure. Class pureness in a sub-sample could mean that the algorithm has found a local maximum in the population. It could also mean the algorithm just doesn't have enough cases to continue the search. If the decision tree learner favors a balanced tree, the region where the learner searches is like a circle. Most search path in the tree will have enough cases to go approximately some fixed distance from the source. If the decision tree learner favors an unbalanced tree, Some search paths will have so many cases that it can go far from the source. And, as a consequence, other searches will have so few cases that must stay close to the source. This unbalanced strategy may help the

hill climber find higher ground.

5 The Decision Tree Learner

The software is written in MacGambit Scheme 3.0, a Lisp dialect. This project includes a simple decision tree learner, $\mathsf{dt}(X)$, which is designed to make it easy to analyze an arbitrary split method X. Furthermore, this learner has the following additional features.

1. When the label counts in a training sample are equal, a *fair majority learner* uses the label of the 0th case to break the tie. To build a leaf, the algorithm uses the fair majority learner. In the event of a tie, this resolution is more fair than always picking one label. Since this research wants to understand the influence of different split methods, it doesn't want the majority learner's tie resolution to have a significant effect on changing the decision tree learner's bias.

2. When building an interior node, this simple learner will only consider *standard tests*, which ask for the value of a given attribute.

3. If a split method returns more than one test and their score is positive, this learner picks one of the tests at random. Since the research wants to understand the influence of different split methods, it doesn't want the learner to have favoritism to any attributes.

4. If the selected tests have a non-positive score, the learner will build a leaf node. A non-positive score indicates that knowing the outcome of a test does not help to predict the label.

5. This decision tree learner has a simple post-pruning method. For any interior node, if all its immediate children are identical, the post pruning method replaces that interior node with one of its immediate children. Here, the pruned tree always computes the same function as the original tree.

Since this research wants to understand the influence of different split methods, it doesn't want the majority learner's post pruning mechanism to have a significant effect on changing the decision tree learner's bias.

6 Estimating Information Gain Scores for Samples

The upcoming predictive method needs a way to take a learning situation and estimate the (possibly inflated) information gain scores for an arbitrary training sample of some fixed length.

Consider the following.

1. An arbitrary non-empty, finite set D.

2. An arbitrary test within D called T.

3. An arbitrary training sample TS.

4. An arbitrary positive integer o.

The *information gain estimate for T under TS* is

$$\mathsf{ige}_o(T, TS) =$$

$$\mathsf{ig}(T, TS) + (1 - \mathsf{min}(1, \frac{s}{o})) * \mathsf{si}(TS),$$

where ig is the information gain function, s is the length of TS, and si is split information.

The integer o should be the number of occurring elements in the situation ($P(d) \neq 0.0$). Note that as s increases to o, $\mathsf{ige}_{(s,o)}(T, TS)$ converges to $\mathsf{ig}(T, TS)$.

7 Building Decision Tree From Situations

Most of the training sample functions that a decision tree uses are not concerned with case ordering. They treat a training sample like a multiset of cases, which maps each case in the domain to a count. Similarly, a concept learning situation defines a case probability distribution, which maps each case in the domain to a probability. It follows that there is a way to transform an arbitrary training sample function into

a corresponding concept learning situation function.

Let $\mathsf{dt}'(\mathsf{ig})$ be a function that maps concept learning situations to decision trees. It is like $\mathsf{dt}(\mathsf{ig})$, except for the following.

1. This function replaces training sample functions with functions for concept learning situations.

2. If the split method selects tests that have a non-positive score, $\mathsf{dt}'(\mathsf{ig})$ will build an interior node.

The second difference is ok, because the following transformation is not interested in this tree's accuracy on unseen elements. The following predictive method (PM) uses $\mathsf{dt}'(\mathsf{ig})$ to characterize the given concept learning situation's target concept.

8 The Information Gain Versus Gain Ratio Predictive Method

The information gain versus gain ratio predictive method PM calls the following recursive function (with embedded comments). This subroutine returns a triple of probabilities, which represents a probability distribution. If the first triple component is the greatest, the PM returns L0_wins. If the second triple component is the highest, the PM returns L1_wins. Otherwise, the method returns tie.

Let *len* be the training sample size parameter.

1. Regarding *len* arbitrary trials of the given situation's case experiment, if the likelihood of seeing a mix of members and non-members in the arbitrary training sample is $< \frac{1}{3}$, return $(0, 0, 1)$.

   Here, this predictive method assumes the split method is rarely invoked. Therefore, both decision tree learners will usually behave like a fair majority learner.

2. Calculate (a, s), the average and standard deviation of the *sort sizes* in the domain, the number of outcomes for each attribute.

3. If $s < \frac{1}{10}$, then do the following. Calculate l, the number of leaves in $\mathsf{dt}'(\mathsf{ig})$'s hypothesis on the situation. Also calculate o the number of occurring domain elements in the situation. If $\frac{o}{l} < 2$ and $\frac{len}{o} \geq \frac{1}{2}$, then return $(0, 1, 0)$. Otherwise, return $(0, 0, 1)$.

   When an attribute has "many values," the largeness of its number of values is with respect to the average number of values of all the attributes in the domain. If $s < \frac{1}{10}$, this subroutine assumes each attribute has approximately the same number of values. I. e., there are no attributes that stands out as having many values.

   If $\frac{o}{l} < 2$ and $\frac{len}{o} \geq \frac{1}{2}$, the predictive method assumes there are many deep paths in the tree for the situation and the training sample is large (with respect to the number of occurring elements). Here, the deep paths

of gain ratio's uneven trees will identify some of the unseen elements. In contrast, information gain may tend to build balanced trees whose paths are too short to accurately identify any collection of elements. Therefore, $\mathsf{dt}(\mathsf{gr})$ wins.

If the second condition is not valid, the information gain split method should select the same tests as the gain ratio split method. This results in a tie.

4. Calculate v, the variance of the information gain scores (with respect to the given situation), for the eligible tests. If $v < \frac{44}{100000}$ then return $(1, 0, 0)$.

   Now, this predictive method will usually judge each learner by how well it picks the current root node. Building a poor root node breaks up the learning problem into sub-problems that are almost as hard as the original problem. Furthermore, the training samples of these sub-problems are smaller.

   If $v < \frac{44}{100000}$, the method assumes all the test have approximately the same information gain scores, when applied to the entire population. Here, gain ratio's uneven trees become a problem. The motivation of the uneven tree is to sacrifice understanding in some areas to become an expert in another area. Unfortunately, there is no area where a long path can achieve a big enough win to compensate for underdeveloping the other paths. Consequently, $\mathsf{dt}(\mathsf{ig})$ outperforms $\mathsf{dt}(\mathsf{gr})$.

5. Calculate r, the ratio between the given training sample length and the number of occurring elements (with non-zero probability) in the given situation. Also, let $d = 1$ if $r < \frac{18}{1000}$, else 2.

Lastly, calculate p, the likelihood of seeing a test with a number of outcomes $< a$ at depth no deeper than d in $\mathsf{dt'}(\mathsf{ig})$'s hypothesis on the situation.

6. If $p \geq \frac{2}{5}$, then do the following.

> *If the preceding condition is valid, then the method assumes that tests with few outcomes can make a significant contribution to improving generalization accuracy. (The closer a node is to the root, the more likely that attribute's value is checked, during hypothesis application.)*

  (a) If the information gain, the gain ratio, and the information gain estimate split methods select the same non-empty test list, and the average sort size in this test list is greater than the average sort size of all the attributes, then do the following. Use the first test to partition the situation, apply this subroutine recursively on each partition element (with the first test removed from the eligible list), and average the results.

> *If this condition is valid, the PM assumes that, when applied to a training sample, the two split methods will pick the same test (which should have many outcomes). So it make sense to take the average of all the predictions on the subproblems.*

  (b) Otherwise, return $(0, 1, 0)$.

> *If this condition is not valid, the method assumes the two split*

methods will not pick the same test. Since information gain's favoritism to attributes with many outcomes may become a detriment, it is more likely to lose.

7. If $p < \frac{4}{25}$, then do the following. If the information gain split method and the information gain estimate split method selected the same test list, then return $(1, 0, 0)$. Otherwise, return $(0, 1, 0)$.

> *If $p < \frac{4}{25}$, the predictive method assumes that tests with few outcomes cannot make a significant contribution to improving generalization accuracy. If the second condition is valid, the method assumes information gain is likely to select the best test and win. If the second condition is not valid, the method assumes information gain does not select the best test. This leads to the construction of a tree with a less than optimal root node, which gives the other split method an opportunity to win.*

8. return $(0, 0, 1)$.

> *If the logic reaches this point, this method gives up and calls it a draw.*

Step 3 is surprising. If there are no attributes with a relatively large number of outcomes, some people might expect a tie between the two learners. The next section presents experiments where the domain has evenly-sized attributes, yet gain ratio wins.

9   Testing

For an arbitrary situation and training sample size, comparing the generalization accuracies of two learn-

10

ers is challenging. In theory, one could calculate the generalization accuracies of the two learners and compare the results. In practice, this strategy is intractable, because the training sample population grows exponentially as a function of the training sample size n.

To get around this problem, a testing procedure was constructed that first generates a random collection of random training sample samples. Then it uses the collection to produce a collection of the difference estimates. Lastly, it takes the difference estimates and calculates the following t test statistic.

$$t \;\; = \;\; \frac{a}{\frac{s}{\sqrt{n}}},$$

where a is the sample mean of the estimates, s is the sample standard deviation of the estimates, and n is the number of estimates. A t test helps statisticians to use estimate statistics to make statistical claims about a population.

The tables in this section contain test examples that support the predictive method's effectiveness with $\mathsf{dt(ig)}$ and $\mathsf{dt(gr)}$. These upcoming tables help to compare the predictive method results with an estimate of the differences in generalization accuracies. In these tables, the first column contains target con-

cepts, which describe the positive label probability vectors. The "training sample sample size" column is the number of random fixed size training samples used to calculate an estimate of the generalization accuracy. In the previously described t test, the likelihood of a type 1 error is set to 0.05. The success of these statistical experiments support the results of PM.

In upcoming tables, let "odd bits on" represent

$$\{(\mathsf{f},\mathsf{f},\mathsf{f},\mathsf{t})\} + \{(\mathsf{f},\mathsf{f},\mathsf{t},\mathsf{f})\} + \{(\mathsf{f},\mathsf{t},\mathsf{f},\mathsf{f})\}+$$
$$\{\{(\mathsf{f},\mathsf{t},\mathsf{t},\mathsf{t})\} + (\mathsf{t},\mathsf{f},\mathsf{f},\mathsf{f})\} + \{(\mathsf{t},\mathsf{f},\mathsf{t},\mathsf{t})\} +$$
$$\{(\mathsf{t},\mathsf{t},\mathsf{f},\mathsf{t})\} + \{(\mathsf{t},\mathsf{t},\mathsf{t},\mathsf{f})\}.$$

This set has all quadruples that have exactly one or three t values.

Table 7 on page 13 presents learning situations where the predictive method declares a dead heat between $\mathsf{dt(ig)}$ and $\mathsf{dt(gr)}$. Since the domain is a bit vector of length 4, there are no attributes with many values. When testing these situations, the null hypothesis was "the difference is zero." Its refutation will support the belief that one of the split methods performs better under this situation.

In the first row, a training sample with both positive and negative cases is likely to occur only 27.58% of the time. So, the PM declares a tie. In the ex-

0	1	2	3	
f	t	f	f	t
f	f	f	f	f
f	t	t	f	f
t	t	f	t	t
t	t	f	t	t
f	f	f	f	f
f	t	t	f	f
f	f	f	t	t
f	t	t	t	t
f	f	t	t	f

Table 9: A sample of the "odd bit on" situation.

periments, the 5 difference estimates are 0. Since the sample standard deviation is 0, the t test statistic is undefined and statisticians cannot perform a t test. This supports the belief that their generalization accuracies are equal.

In the remaining rows, a training sample with both positive and negative cases is likely to occur more than a third of the time. But all the attributes have the same number of values.

For domains that have no attributes with many outcomes, there is a small class of situations where the two learners achieve a non-zero difference in generalization accuracy. Some of these situations are presented in table 8.

Example 9.1 *To understand why information gain loses to gain ratio, let's examine the "odd bits on" situation, when the training sample size is 10.*

Look at the training sample described in table 9. For this training sample, the difference in generalization accuracy is comparatively large. Examining this training sample helps to understand how the split method affects the learner's bias.

When applied to table 9, dt(ig) and dt(gr) builds

decision trees that represent the hypotheses

$$\{(*, t, f, f)\} + \{(*, f, f, t)\} + \{(*, t, *, t)\}$$

and

$$\{(f, t, f, f)\} + \{(f, f, f, t)\} + \{(f, t, *, t)\} + \{(t, *, *, *)\}$$

respectively. Table 9, has a size of 10 and three duplicate elements: (f, f, f, f), (f, t, t, f), and (t, t, f, t). So the concept learning problem has seven seen members and there are nine unseen members in the domain. Information gain's hypothesis can only classify one unseen domain member correctly, (t, f, t, f) (it is a target concept non-member). Therefore, the generalization accuracy with respect to the given sample is $\frac{1}{9}$. Gain ratio's hypothesis is a little better; it can correctly classify three unseen members, (t, f, f, f), (t, f, t, t), and (t, t, f, t). This achieves a generalization accuracy of $\frac{1}{3}$.

Regarding table 10 on page 13, the information split method chooses attribute 3 as the root node. This partitions the domain into two roughly equal-size training samples. dt(ig) goes on to build a balanced tree. The gain ratio split method chooses attribute 0 as the root node. This partitions the domain into a small, pure training sample and a large training sample. The small, pure training sample becomes the only tree path of length 1. dt(gr) ultimately builds an uneven tree.

In this situation, most decision tree generalizations misclassify more than half the unseen elements that they cover. Since gain ratio has favoritism to unbalanced trees, it is able to create two paths of length 4. The small partition elements $\{(f, t, f, f)$

target concept	training sample size	training sample sample size	gen accuracy est size	diff	std dev.	alpha
(t, t, t, t)	5	10	5	0	0	Ω
$(*, f, *, f)$	10	20	10	0.002	0.006	0.343
$(*, f, *, *)+$	10	10	5	0.003	0.007	0.191
$(*, *, *, f)$						
odd bits on	2	25	12	0	0	Ω
odd bits on	6	25	12	0	0	Ω

Table 7: (Information Gain Ties Gain Ratio) Situations where $D = \{f, t\} \times \{f, t\} \times \{f, t\} \times \{f, t\}$ and P is equally likely.

target concept	training sample size	training sample sample size	gen accuracy est size	diff	std dev.	alpha
odd bits on	8	25	12	-0.001	0.002	0.020
odd bits on	10	25	12	-0.006	0.008	0.017
odd bits on	12	25	12	-0.005	0.007	0.009

Table 8: (Information Gain Loses to Gain Ratio) Situations where $D = \{f, t\} \times \{f, t\} \times \{f, t\} \times \{f, t\}$ and P is equally likely.

attribute	information gain	split info	gain ratio
0	0.236	0.722	0.328
1	0.125	n/a	n/a
2	0.125	n/a	n/a
3	0.278	1.0	0.278

Table 10: Information gain and gain ratio scores for table 9

and $\{(f,f,f,t)\}$ characterize these long paths. For such paths, there is no generalization. Furthermore, it also creates a path of length 1, which is characterized by the large partition element $\{(t,*,*,*)\}$. This rule classifies half of the unseen elements it matches correctly.

In contrast, information gain's tree creates paths of length less than 4. This builds medium size partition elements that generalize in a manner that misclassify too many unseen elements.

Now this section examines situations where the domain has attributes with many values. For every row in the following tables, the likelihood that an arbitrary training sample of the specified length has mixed cases is above $\frac{1}{3}$.

Two domains were selected whose first and third attributes have many outcomes and the other attributes have few outcomes.

Table 11 presents situations where the predictive method declares dt(gr) the winner. Note that the simple target concept of these situations always includes an attribute with few outcomes. When compared to the gain ratio, the information gain split method has a harder time selecting that attribute for the tree. When PM builds a decision tree from the situation, this attribute is likely to occur in the tree at a shallow depth. Consequently, when given a training sample, dt(ig) has a harder time developing a hypothesis that is as effective as the hypothesis of its competition.

Table 12 and 13 present situations where the predictive method declares dt(ig) the winner. Here, the simple target concepts do not include an attribute with few outcomes. When compared to its brethren, dt(ig) is less hesitant to build an effective hypothesis, composed of only attributes with many outcomes. So, its generalization accuracy is better.

Table 14 presents a situation where the given PM declares a tie. Here, the simple target concept has an attribute with few outcomes. However, the concept also has two attributes with many outcomes. When building a decision tree from the situation, this attributes is likely to occur in the tree at a deep depth. Consequently, the overall difference in generalization accuracy between dt(ig) and dt(gr) is insignificant.

It was challenging to apply the real world problems in the Cal-Irvine Machine Learning repository to this research. This research wants to compare the generalization accuracy of two decision tree learners on arbitrary concept learning situations. Unfortunately, the Cal-Irvine repository contains concept learning problems. While concept learning problems often provide

target concept	training sample size	training sample sample size	gen accuracy est size	diff	std dev.	alpha
(*, 1, *, 1)	5	10	5	-0.015	0.014	0.034
(*, 1, *, 1)	10	10	5	-0.092	0.075	0.027
(*, 1, *, 1)	20	10	5	-0.056	0.029	0.0063
({0, 1, 2, 3}, 0, *, *)	10	10	5	-0.046	0.032	0.02
({0, 1, 2, 3}, 0, {0, 1, 2, 3}, *)	10	10	5	-0.038	0.031	0.017
({0, 1}, 0, *, *)	10	10	5	-0.025	0.020	0.026

Table 11: (Information Gain Loses to Gain Ratio) Situations where $D = \{0, 1, 2, 3, 4\} \times \{0, 1\} \times \{0, 1, 2, 3, 4\} \times \{0, 1\}$ and P is equally likely.

target concept	training sample size	training sample sample size	gen accuracy est size	diff	std dev.	alpha
({0, 1, 2, 3}, *, {0, 1, 2, 3}, *)	10	10	5	0.008	0.0073	0.0354
({0, 1, 2, 3}, *, {0, 1, 2, 3}, *)	50	10	5	0.010	0.010	0.039
({0, 1, 2, 3}, *, {0, 1, 2}, *)	10	10	10	0.010	0.018	0.049
({0, 1, 2}, *, {0, 1, 2}, *)	10	10	10	0.0179	0.0094	0.040

Table 12: (Information Gain Beats Gain Ratio) Situations where $D = \{0, 1, 2, 3, 4\} \times \{0, 1\} \times \{0, 1, 2, 3, 4\} \times \{0, 1\}$ and P is equally likely.

target concept	training sample size	training sample sample size	gen accuracy est size	diff	std dev.	alpha
({0, 1, 2}, *, {0, 1}, *, *)	10	10	10	0.010	0.0180	0.037
(0, *, 0, *, *)	25	25	10	0.005	0.005	0.0054

Table 13: (Information Gain Beats Gain Ratio) Situations where $D = \{0, 1, 2, 3, 4\} \times \{0, 1\} \times \{0, 1, 2, 3, 4\} \times \{0, 1\} \times \{0, 1\}$ and P is equally likely.

target concept	training sample size	training sample sample size	gen accuracy est size	diff	std dev.	alpha
({0, 1}, 0, {0, 1}, *)	10	10	10	0.002	0.005	0.247

Table 14: (Information Gain Ties Gain Ratio) Situations where $D = \{0, 1, 2, 3, 4\} \times \{0, 1\} \times \{0, 1, 2, 3, 4\} \times \{0, 1\}$ and P is equally likely.

a finite domain, they often don't supply a domain probability, or a positive label probability vector.

Fortunately, it is possible to transform a training sample into a concept learning situation. For a training sample, the derived situation's domain probability distribution is the distribution of domain elements in the sample. The ith component of the derived positive label probability vector is the percentage of cases in the sample that have 1 as their label, when we know the domain element of the case is the ith element.

To take advantage of these real world training samples, additional test procedures were written that treat a training sample like a concept learning situation.

Example 9.2 *Recall example 3.1, the concept learning problem derived from a problem in the Cal-Irvine Machine Learning Repository. Table 2 presents the problem domain.*

This example predicts the winner when the sample sizes are 10 and 40.

For either sample size, the application of dt′(ig) *to the population produces a tree, where the likelihood of seeing a test with few values near the root is 0. And while the application of the information gain split method on the population picks attribute 3, the application of the information gain estimate split method picks attribute 0. It follows that the PM picks* dt(gr) *as the winner.*

*When testing this situation for training sample sizes 10 and 40, the training sample sample size was set to 7 and the generalization estimate sample was set to 3. The null hypothesis is "*dt(ig) *wins or ties".*

For the stochastic experiment with sample length 10, the average sample difference is -0.035 and the sample standard deviation is 0.018. The alpha score

is 0.038, which supports gain ratio's victory.

For the stochastic experiment with sample length 40, the average sample difference is -0.030 and the sample standard deviation is 0.017. The alpha score is 0.045, which also supports gain ratio's victory.

Table 15 graphs the three difference estimates, when the training sample length is 40. Gain ratio outperforms information gain in all the individual experiments.

Example 9.3 *Consider changing the problem in example 3.1, so it identifies the women that use long term contraceptives.*

Now this example predicts the winner when the sample size is 10.

Since the variance between the information gain scores is so small, the PM picks dt(ig) *as the winner.*

*When testing this situation for a training sample of length 10, the training sample sample size was set to 10 and the generalization estimate sample was set to 5. the null hypothesis is "*dt(gr) *wins or ties".*

In the stochastic experiment, the average sample difference is 0.022 and the sample standard deviation is 0.011. The alpha score is 0.006, which refutes the null hypothesis.

Table 16 graphs the five difference estimates, when the training sample length is 10. Here, gain ratio loses to information gain in all the individual experiments.

10 Conclusions

By using generalization accuracy as a metric and a predictive method as a framework, we have developed a novel way to compare the bias of different learners. Here, we have developed a predictive method that correctly predicts the better learner for many situta-
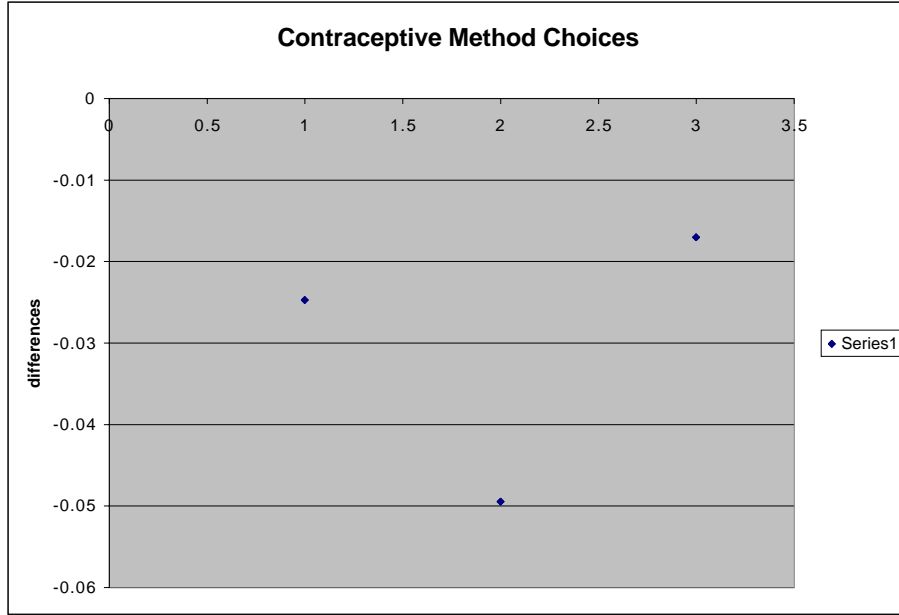
Table 15: (Information Gain Loses to Gain Ratio) Contraceptive method choice differences, where $len = 40$.
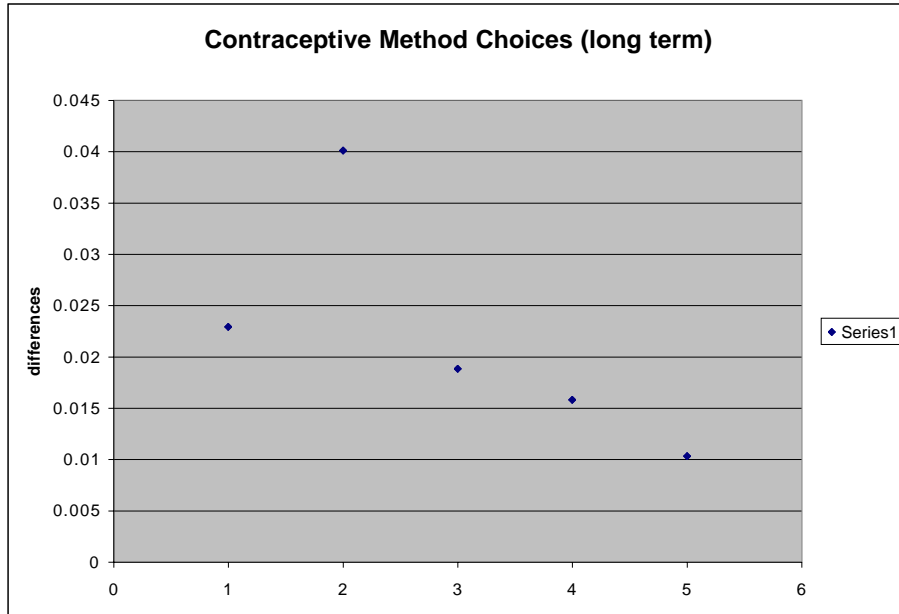

Table 16: (Information Gain Beats Gain Ratio) Contraceptive method choice differences, where $len = 10$.

tions. Furthermore, we have shown that the belief that gain ratio normalizes information gains unjustified favoritism is a simplification. By examining concept learning situations, we can see that gain ratio also encourages the development of uneven trees.

11   Related Research

Schaffer studied a decision tree learner that uses the following split method [7]. Consider the following.

1. An arbitrary input space DOM.

2. An arbitrary test within DOM t.

3. An arbitrary training sample ts.

The *information loss for t under ts* is

$$\mathsf{il}(t, ts) \quad = \quad -\mathsf{ig}(t, ts)$$

Based on intuition, a decision tree learner that uses the il split method will have lower generalization accuracy than decision tree learners that use ig or gr. Schaffer's article shows that this learner performs performs better than the conventional learners for some concept learning situations that occur in the "real world."

Some believe the no strict improvement corollary is practically irrelevant; there are learners that are generally better than other learners on "real world" problems. Schaffer wanted to show that there are real world problems that need to be solved with unintuitively sound learners.

This research analyzes learners that use split methods that are more intuitively sound than il. This research shows how scientists can use generalization accuracy to get a better understanding of the strengths and weaknesses of intuitively sound learners.

Some research [5] has a philosophy that diverges from Schaffer's in the following way.

1. Schaffer's law sums up the generalization performance for a collection of situations, as the training samples vary. The *expected generalization performance for a situation and training sample size* is a weighted sum, based on the occurrence likelihood of positive label probability vectors. The new philosophy claims that expected generalization performance is more practically useful.

2. Schaffer's article says that researchers should try to understand the bias of learners. The new philosophy claims that researchers should try to understand the likelihood (in practice) of seeing a certain situation.

The second point supports the existence of a learner that have maximal expected generalization performance on the hypothetical collection of "real world" situations. As a benefit, a machine learning researcher can effectively ignore the consequences of the conservation law. Though the conservation law warns that the sum of any learner's generalization performance over all positive label probability vectors is 0, a good learner will have positive expected generalization performance in practice.

There are several authors that support this alternative philosophy [2]. These authors study bias evaluation and selection; this area of research concentrates on composite (or multi-strategy) learners. When a composite learner begins to analyze training samples, it uses a particular learning paradigm. If the current paradigm seems to perform poorly on the given concept learning problem, the composite learner changes the paradigm. Again, these learners should have positive expected generalization performance in practice.

In contrast, this research is more supportive of Schaffer's philosophy. This dissertation concentrates on understanding the bias of a several decision tree learners. It makes no assumptions about a learner's expected generalization performance for the class of "real world" situations.

There are several authors that recommend using Receiver Operating Characteristic (ROC) curve analysis as a metric for comparing learners [3]. They criticize the use of accuracy as a measure of performance for real world problems. And while accuracy assumes equal misclassification cost, these cost are often widely different in real world problems. While some might suggest adjusting the input space distribution to reflect the difference in cost, these authors claim that misclassification costs are difficult to determine. By studying ROC curves, they try to find models that are dominant under any cost adjustments.

References

[1] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[2] Marie desJardin and Diana F. Gordon, editors. *Machine Learning*, volume 20. Kluwer Academic Publishers, Boston, Massachussetts, July/August 1995.

[3] Tom Fawcett Foster Provost and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the*

Fifteenth International Conference on Machine Learning (ICML-98), 1998.

[4] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, Inc., San Mateo, California, 1993.

[5] R. Bharat Rao, Diana Gordon, and William Spears. For every generalization action, is there really an equal and opposite reaction? analysis of the conservation law for generalization performance. In Armand Prieditis and Stuart Russell, editors, *Machine Learning*, volume 12, 1995.

[6] Cullen Schaffer. A conservation law for generalization performance. In William W. Cohen and Haym Hirsh, editors, *Machine Learning*, volume 11, 1994.

[7] Cullen Schaffer. Conservation of generalization: A case study (draft). posted on World Wide Net, February 1995.