Averaging Over Decision Stumps

Jonathan J. Oliver and David Hand

Department of Statistics, Open University
Walton Hall, Milton Keynes, MK7 6AA, UK

Abstract. In this paper, we examine a minimum encoding approach to the inference of decision stumps. We then examine averaging over decision stumps as a method of generating probability estimates at the leaves of decision trees.

1 Introduction

We discuss the supervised learning problem with particular emphasis on small data sets. Supervised learning involves constructing a rule from a training set of examples, where each example has a set of measurements and a class associated with it. We then use the constructed rule to make predictions of the class for future examples.

It is our experience that very simple rules may produce quite high predictive accuracy on many data sets. For example, a Bayes classifier [14] which assumes that the attributes are statistically independent, can perform well even in those cases when the independence assumption is clearly inappropriate.

In this paper, we focus on classification rules that take the form of decision trees [3, 15]. It has been found that decision tree like rules, that only consider one or two attributes, can perform surprisingly well [11, 12, 25]. In particular, Holte [11] and Iba and Langley [12] considered rules that only considered one attribute. These simple rules are in effect severely pruned decision trees and have been termed *decision stumps* [12].

The majority of machine learning systems determine a model to explain some data by considering a range of models, and selecting a single model that satisfies some criterion. Some authors argue that in some circumstances, a set of models can explain data better than a single model [7]. Within the field of decision trees, pruning is equivalent to selecting a single tree, which can then be used for making predictions for future examples. Tree averaging [2, 4, 5, 8, 10, 13, 16] involves making predictions using a set of trees.

This paper is organised in the following way. Section 2 defines decision stumps. In Sections 3 and 4, we examine the relationship between pruning and tree averaging. Section 5 demonstrates how MML can be used to assign weights for tree averaging. In Section 7, we replace each leaf of a decision tree with a weighted set of decision stumps. In some cases this improves predictive accuracy of a decision tree, without the computational cost associated with option trees.

2 Decision Stumps

We identify height 1 (a single leaf) and height 2 decision trees as decision stumps. Consider the LED data set described by Breiman et al. [3]. This data set has 10 classes representing whether the LED is showing the digits 0-9. The LED has 7

elements (A1 — A7), as shown in the key in Figure 1, each of which has a 10% chance of being faulty. Figure 1 depicts the 8 decision stumps for the LED data set and shows where a non faulty display would categorise each digit.

Figure 1: Decision stumps for the LED data set

When we consider decision stumps for data sets with continuous attributes, then we may construct multiple decision stumps for the one attribute. We restrict ourselves in this paper, to one decision stump for each continuous attribute, by selecting the cut point which minimises a minimum encoding metric.

3 Pruning

Constructing a decision tree is traditionally performed in 2 phases, a *Growing Phase*, and a *Pruning Phase*. The Growing Phase grows a *complete* decision tree by recursively splitting each leaf until each leaf is either class pure, or considered too small to split. The Pruning Phase takes as input the complete tree T . Its task is to select a single “best” tree from the *set of pruned trees* of T , $\{PT_1, PT_2, \dots, PT_n\}$. The set of pruned trees contains each tree that has 0 or more decision nodes transformed into leaves. For example, the set of pruned trees for the tree T in Figure 2 $= \{PT_1, PT_2, \dots, PT_{13}\}$.

The measures used for pruning are typically statistical estimates of the error rate, and pruning involves finding the pruned tree that minimises this estimate. The estimates of error are typically calculated by *cross validation* [3].

4 Tree Averaging

Tree averaging or *smoothing* is a generalization of pruning that has recently been receiving more attention. Buntine [4, 5], Kwok and Carter [13], Hastie and

4.1 Option Trees

Buntine [4, 5] took the idea of tree averaging one step further. Tree averaging considers the set of pruned trees of a single tree. *Option trees* represent a large set of trees, and average over the set of pruned trees for each tree in this set. The result of this is that option trees use more computational resources than traditional decision trees. However, this is offset by the improved predictive ability of option trees [4, 5].

5 MML Inference

The minimum encoding learning paradigms were developed during the 1960s and 1970s. There are two schools of minimum encoding inference: Minimum Message Length (MML) — developed by Wallace et al. [9, 22, 23, 24], and Minimum Description Length (MDL) — developed by Rissanen [17, 18, 19, 20]. These learning paradigms are applicable to a wide range of problems, since they were developed to “fit models to data”. An introduction is given by Georgeff and Wallace [9].

These approaches turn the problem of finding a good hypothesis into the problem of minimizing an encoding of the data. Given a hypothesis, we can calculate the length of encoding the hypothesis and the data given the proposed hypothesis. Thus, these approaches offer a clear answer to the bias problem, i.e., searching for the hypothesis that has the smallest encoding of the hypothesis and data.

Decision trees can be regarded as a theory about a domain, mapping objects to classes. We define an *explanation* of the class vector as a message encoded as a binary string and consisting of two parts. The first part states the decision tree, T , and the second part states the class vector using a code that gives the minimum expected message length using T .

Quinlan and Rivest [17] and Wallace and Patrick [24] have presented encoding scheme for decision trees. In this paper, we use a slightly modified form of the code used by Wallace and Patrick [24]. We note that an encoding of decision stumps should use the fact that we are willing to split only the root node.

5.1 Averaging over Theories Using MML

When given a set of trees, $\{T_1, T_2, \dots, T_n\}$, about some data, we may calculate the message length, ML_i of the explanation provided by each tree, T_i . We may establish a posterior probability distribution over trees:

$$Prob(T_i|D) \propto Prob(D \& T_i) = 2^{-ML_i}$$

We may use the posterior distribution over the theories, in 2 ways: (1) we may select that tree with the highest posterior probability, or (2) we may use the posterior distribution over trees to estimate expectations for new objects, as done by Allison et al. [1] and Buntine [4, 5]. We determine the expected probability of object x

belonging to class c as:

$$E(\text{Prob}(\text{Class} = c|x)) = \sum_{T_i \in \text{models}} \text{Prob}(T_i|D) \times \text{Prob}(\text{Class} = c|x, D, T_i) \quad (1)$$

Equation 1 can be rewritten as

$$E(\text{Prob}(\text{Class} = c|x)) = \sum_{T_i \in \text{models}} w_i \times \text{Prob}(\text{Class} = c|x, D, T_i) \quad (2)$$

where each tree has weight $w_i = 2^{-ML_i}$. In practical applications there is an infeasible number of decision trees to calculate this sum. Tree averaging and option trees restrict the *models* set, and attempt to approximate this sum.

5.2 Weighting Schemes for Decision Stumps

We consider 3 weighting options for decision stumps:

1. Weighting each stump equally. Hence if there are N available decision stumps, then we weight each stump with $w_i = \frac{1}{N}$
2. Assigning weight 1.0 to the decision stump with minimum message length, and weight 0.0 to the other decision stumps.
3. Weighting each stump according to the posterior implied by the MML encoding scheme. If decision stump i has message length ML_i then we weight stump i with weight $w_i = 2^{-ML_i}$

To classify a test object, O we construct the expected frequency distribution over classes, and assign the most frequent class to object O .

$$E(\text{Freq}(\text{class}_j | O)) = \sum_{i=1}^{n\text{stumps}} w_i \times \text{Freq}(\text{class}_j | O, \text{stump}_i)$$

6 Empirical Evaluation of Decision Stumps

We consider the following data sets ¹: Geographic (described by Wallace and Patrick [24]), Glass (described by Buntine [6]) LED (described by Breiman et al. [3]), Mushroom (described by Schlimmer and Granger [21]) and Pole (described by Buntine [6]). A summary of some properties of these data sets is given in Table 1. The base accuracy is the proportion of cases that have the most common class.

6.1 Results

We implemented the 3 weighting options for decision stumps described in Section 5.2. Table 2 gives the percentage accuracy for decision trees [24] and the 3 decision stump schemes on the data sets summarised in Table 1. These results were averaged over 10 runs. For each run a training set was randomly created, and the accuracy was calculated using the remainder of the data set as the test set.

¹The majority of these data sets come from the UCI repository.

Data Set	Cont Attr	Binary Attr	Multi val. Attr	Total Attr	No. Class	Size	Base Acc.
Geographic	8	-	-	8	4	106	37.7 %
Glass	9	-	-	9	6	214	35.5 %
Heart Disease	5	3	5	13	2	303	54.1 %
LED	-	7	-	7	10	3000	10.0 %
Mushroom	-	4	18	22	2	8124	51.8 %
Pole	4	-	-	4	2	1847	51.0 %

Table 1: A summary of the data sets considered

		Decision Tree		Equal Weights		Best Stump		MML Weights	
Data Set	Items	Accuracy %		Accuracy %		Accuracy %		Accuracy %	
Geo	10	33.3	± 8.0	42.1	± 8.7	33.3	± 8.0	36.9	± 3.9
	20	36.3	± 11.2	39.2	± 8.3	36.3	± 11.2	39.1	± 8.2
	30	42.0	± 7.3	41.7	± 7.7	41.7	± 6.3	45.5	± 10.2
	40	53.9	± 11.2	40.5	± 7.4	46.1	± 7.5	49.5	± 10.7
p-value		0.80		0.58		0.057		—	
Glass	10	29.0	± 10.6	34.4	± 6.5	29.0	± 10.6	29.0	± 10.6
	20	34.7	± 3.4	34.6	± 5.4	34.7	± 3.4	35.3	± 4.7
	30	39.7	± 6.0	40.0	± 9.1	38.6	± 6.3	42.0	± 6.3
	40	41.4	± 3.7	36.1	± 4.4	41.3	± 3.8	43.0	± 4.4
p-value		0.21		0.12		0.21		—	
Heart Disease	10	51.7	± 6.5	62.4	± 13.3	53.6	± 9.6	62.4	± 15.2
	20	58.0	± 11.5	66.8	± 13.6	63.4	± 10.7	66.6	± 10.8
	30	68.8	± 10.5	67.4	± 10.5	73.0	± 3.2	75.7	± 3.0
	40	71.9	± 3.2	70.0	± 12.0	71.9	± 3.2	73.3	± 3.1
p-value		3.0×10^{-6}		1.00		1.5×10^{-5}		—	
LED	10	10.2	± 0.4	19.4	± 7.6	10.2	± 0.4	18.0	± 9.5
	20	21.9	± 5.0	20.2	± 5.8	18.4	± 0.5	22.1	± 5.3
	30	30.9	± 9.6	20.5	± 3.1	18.1	± 0.6	24.1	± 5.6
	40	43.0	± 9.6	23.5	± 7.2	18.1	± 0.6	24.1	± 6.7
p-value		NEG 0.62		1.0		3.7×10^{-9}		—	
Mushroom	10	67.3	± 17.7	72.6	± 15.3	67.3	± 17.7	77.5	± 16.2
	20	89.1	± 5.4	77.1	± 17.0	89.1	± 5.4	89.9	± 6.0
	30	95.5	± 2.7	81.4	± 13.1	94.9	± 3.8	96.0	± 2.1
	40	95.6	± 3.8	80.1	± 17.2	95.0	± 4.7	95.2	± 5.0
p-value		0.0013		5.4×10^{-9}		0.00054		—	
Pole	10	60.5	± 17.2	63.9	± 10.8	60.5	± 17.2	64.4	± 17.7
	20	70.5	± 18.2	63.8	± 9.6	70.5	± 18.2	76.4	± 14.6
	30	80.5	± 11.5	64.3	± 12.9	80.5	± 11.5	80.5	± 11.3
	40	85.1	± 0.2	67.0	± 15.0	85.1	± 0.2	85.1	± 0.2
p-value		0.031		0.0019		0.031		—	

Table 2: The accuracy of decision trees and decision stumps on some data sets

In Table 2, the standard deviation of the accuracy is indicated after the “ $\pm$ ” symbol. The p-values give the statistical significance of the 40 simulations performed for each domain when each method is compared with the MML weighted stump

scheme. The p-value for the LED domain is labelled “NEG” since decision trees outperformed MML weighted stumps (but not significantly). The method used to obtain these p-values is described in the next section.

6.2 Analysing the Simulation Outcomes

For a given data set, to compare the accuracy of two classification rules we need to take into account the fact that the ten observations at each training set size are matched (the same ten training samples were used to design both classifiers). This means that the standard deviations of the differences are in fact much smaller than those derived from the standard deviations given in Table 2. We could perform matched paired t-tests (or nonparametric equivalents) at each training set size but, with only ten runs at each size, these tests would not be very powerful. Attempts to combine the four training set sizes into a single analysis (by analysing the forty results in a single paired comparison test or by using analysis of variance with training set size as a factor) are complicated by the fact that the variances in the four training set size groups are different. This invalidates paired comparison tests because it breaks the iid assumption and it invalidates the analysis of variance because it breaks the homoscedasticity assumption. A proper formal analysis would involve an appropriately weighted analysis of variance. To avoid such complications in this paper, we have simply computed the signs of each of the forty differences and conducted a sign test for those differences which are non-zero. (Even this is not perfect since it makes the latent assumption that the medians of the population distributions of differences do not change sign over the range of training set sample sizes we have analysed.) We did this for each data set (six) and for the comparison of the MML weighted decision stump classifier with each of the other classifiers (three). This results in 18 tests in all.

At the 5% level you expect approximately one of these tests to be significant by chance alone, if the null hypothesis of no difference between classifiers is true. We, however, have obtained 9 out of 18 showing a significant difference favouring the MML weighted scheme.

7 Fanned Decision Trees

The decision stump scheme has another possible use. Traditional decision trees classify a new object using the class distribution at the leaf associated with that object. We can construct all the stumps at each leaf of a decision tree as shown in Figure 3. We term such a structure as a *fanned decision tree*. We may classify a new object, O , using a fanned decision tree by identifying the leaf, L , that we would associate with O , and classifying O using the decision stumps associated with L .

A fanned decision tree is a restricted option tree [4, 5]. The approach we take in this paper has some advantages over option trees:

1. Option trees take up to a magnitude more space and time than decision trees. The additional computational complexity with fanned decision trees is virtually insignificant.
2. Option trees requires considerable skill to set the search parameters.

4 Binary Attributes (A, B, C, D)

Figure 3: A fanned decision tree

We implemented fanned decision trees and repeated the empirical evaluation performed in Section 6. Table 3 gives the percentage accuracy for MML decision trees [24], fanned decision trees and MML weighted decision stumps. The p-values give the statistical significance of the 60 (or 50 in the Geographic domain) simulations performed for each when each method is compared with fanned decision trees. In five out of the six domains, fanned trees performed significantly better than decision trees. The comparison of fanned trees and MML weighted stumps was inconclusive.

8 Analysis of Results and Conclusions

The decision stump scheme described is limited to very small training sets and decision trees (and other classification schemes) will outperform decision stumps with larger training sets. In some cases (such as rare medical conditions) we may only have very small data sets available.

The results in Table 2 and Table 3 lead us to the following conclusions:

- The MML weighted stumps classifier outperformed the equally weighted stumps and the best stump classifiers.
- Over the data sets considered, fanned decision trees outperformed decision trees. Fanned trees significantly outperformed MML weighted stumps in 3 domains, but in the Mushroom domain MML weighted stumps significantly outperformed fanned trees. Of course, these results should be interpreted in the light of the relatively large number of significance tests that have been conducted. In general, fanned trees were similar to MML weighted stumps for small training sets (< 50 items say), and then slightly superior to decision trees for larger training sets.
- Once there was sufficient data for decision trees to grow beyond height 2,

		Tree		Fanned Tree		MML Stump	
Data Set	Items	Accuracy %		Accuracy %		Accuracy %	
Geo	10	33.3	±8.0	36.9	±3.9	36.9	±3.9
	20	36.3	±11.2	41.2	±11.8	39.1	±8.2
	30	42.0	±7.3	47.5	±10.7	45.5	±10.2
	40	53.9	±11.2	57.7	±10.3	49.5	±10.7
	50	60.4	±13.8	60.9	±14.5	50.2	±13.8
p-value		0.00012		—		0.00028	
Glass	10	29.0	±10.6	29.0	±10.6	29.0	±10.6
	20	34.7	±3.4	35.4	±4.9	35.3	±4.7
	30	39.7	±6.0	42.8	±8.7	42.0	±6.3
	40	41.4	±3.7	42.6	±2.1	43.0	±4.4
	50	43.6	±3.4	47.6	±8.0	43.5	±3.5
p-value	100	56.9	±9.6	57.9	±10.2	40.9	±3.3
		0.0013		—		0.0046	
Heart Disease	10	51.7	±6.5	61.4	±14.5	62.4	±15.2
	20	58.0	±11.5	65.7	±10.2	66.6	±10.8
	30	68.8	±10.5	74.4	±2.6	75.7	±3.0
	40	71.9	±3.2	72.2	±2.9	73.3	±3.1
	50	73.3	±2.3	73.3	±2.3	74.7	±1.9
p-value	100	73.5	±2.3	74.2	±1.7	71.9	±4.6
		9.5×10^{-7}		—		NEG 0.071	
LED	10	10.2	±0.4	18.0	±9.5	18.0	±9.5
	20	21.9	±5.0	27.9	±7.8	22.1	±5.3
	30	30.9	±9.6	34.4	±9.0	24.1	±5.6
	40	43.0	±9.6	43.7	±10.4	24.1	±6.7
	50	45.8	±6.1	47.0	±5.6	22.5	±6.4
p-value	100	56.8	±6.8	57.6	±6.2	18.5	±2.0
		6.7×10^{-8}		—		1.0×10^{-7}	
Mushroom	10	67.3	±17.7	77.8	±16.3	77.5	±16.2
	20	89.1	±5.4	89.1	±5.4	89.9	±6.0
	30	95.5	±2.7	95.5	±2.7	96.0	±2.1
	40	95.6	±3.8	95.6	±3.8	95.2	±5.0
	50	96.7	±2.3	96.7	±2.3	97.1	±2.4
p-value	100	98.4	±0.5	98.4	±0.5	98.5	±0.1
		0.25		—		$\text{NEG } 3.8 \times 10^{-5}$	
Pole	10	60.5	±17.2	64.4	±17.7	64.4	±17.7
	20	70.5	±18.2	76.4	±14.6	76.4	±14.6
	30	80.5	±11.5	80.5	±11.3	80.5	±11.3
	40	85.1	±0.2	85.1	±0.2	85.1	±0.2
	50	83.7	±3.1	83.7	±3.1	84.1	±3.0
p-value	100	84.7	±0.9	84.7	±0.9	84.7	±0.9
		0.031		—		1.0	

Table 3: The accuracy of decision trees, fanned trees, and decision stumps

decision trees can outperform decision stumps. However, in half of the domains considered the behaviour between stumps and trees was almost identical.

9 Acknowledgments

We would like to thank the anonymous referees, Andrew Webb and Dave Signorini for their helpful comments. This work has been carried out with the support of the Defence Research Agency, Malvern.

References

- [1] L. Allison, C.S. Wallace, and C.N. Yee. Finite-state models in the alignment of macromolecules. *Journal of Molecular Evolution*, 35:77–89, 1992.
- [2] L.R. Bahl, P.F. Brown, P.V. deSouza, and R.L. Mercer. A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37:1001–1008, 1989.
- [3] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984.
- [4] W.L. Buntine. *A Theory of Learning Classification Rules*. PhD thesis, School of Computing Science in the University of Technology, Sydney, February 1990.
- [5] W.L. Buntine. Learning classification trees. *Statistics and Computing*, 2:63–73, 1992.
- [6] W.L. Buntine and T. Niblett. A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 8:75–85, 1992.
- [7] P. Cheeseman. In defense of probability. In *Proceedings of IJCAI-85*, pages 1002–1009, 1985.
- [8] L.A. Clark and D. Pregibon. Tree-based models. In J.M. Chambers and T.J. Hastie, editors, *Statistical Models in S*, pages 377–420. Wadsworth and Brooks, California, 1992.
- [9] M.P. Georgeff and C.S. Wallace. A general criterion for inductive inference. In *Proceedings of the 6th European Conference on Artificial Intelligence*, pages 473–482, 1984.
- [10] T. Hastie and D. Pregibon. Shrinking trees. Technical report, AT&T Bell Laboratories, Murray Hill, New Jersey 07974, USA, 1990.
- [11] R.C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993.
- [12] W.F. Iba and P. Langley. Induction of one-level decision trees. In *Machine Learning: Proceedings of the Ninth International Workshop*, pages 233–240, 1992.

- [13] S.W. Kwok and C. Carter. Multiple decision trees. In R.D. Schachter, T.S. Levitt, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 4*, pages 327–335. Elsevier Science, Amsterdam, 1990.
- [14] P. Langley, W.F. Iba, and K. Thompson. An analysis of bayesian classifiers. In *Proceedings of AAAI-92*, pages 223–228, 1992.
- [15] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.
- [16] J.R. Quinlan. Learning with continuous classes. In A. Adams and L. Sterling, editors, *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, pages 343–348. World Scientific, Singapore, 1992.
- [17] J.R. Quinlan and R.L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248, 1989.
- [18] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11:416–431, 1983.
- [19] J. Rissanen. Stochastic complexity. *Royal Statistical Society Journal*, 49:223–239, 1987.
- [20] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, Singapore, 1989.
- [21] J. Schlimmer and R. Granger. Incremental learning from noisy data. *Machine Learning*, 1:317–354, 1986.
- [22] C.S. Wallace and D.M. Boulton. An information measure for classification. *Computer Journal*, 11:185–194, 1968.
- [23] C.S. Wallace and P.R. Freeman. Estimation and inference by compact coding. *Royal Statistical Society Journal*, 49:240–252, 1987.
- [24] C.S. Wallace and J.D. Patrick. Coding decision trees. *Machine Learning*, 11:7–22, 1993.
- [25] S.M. Weiss and I. Kapouleas. An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. In *Proceedings of IJCAI-89*, pages 781–787, 1989.