Hybrid Decision Tree Learners with Alternative Leaf Classifiers: An Empirical Study

Alexander K. Seewald[1], Johann Petrak[1], Gerhard Widmer[1,2]
[1] Austrian Research Institute for Artificial Intelligence, Schottengasse 3, A-1010 Vienna,
[2] Department of Medical Cybernetics and Artificial Intelligence, University of Vienna, Austria
{alexsee,johann,gerhard}@ai.univie.ac.at

Abstract

There has been surprisingly little research so far that systematically investigated the possibility of constructing hybrid learning algorithms by simple local modifications to decision tree learners. In this paper we analyze three variants of a C4.5-style learner, introducing alternative leaf models (Naive Bayes, IB1, and multi-response linear regression, respectively) which can replace the original C4.5 leaf nodes during reduced error post-pruning. We empirically show that these simple modifications can improve upon the performance of the original decision tree algorithm and even upon both constituent algorithms. We see this as a step towards the construction of learners that locally optimize their bias for different regions of the instance space.

Introduction

Tree-based learning methods are widely used for machine learning and data mining applications. These methods have a long tradition and are commonly known since the works of (Breiman et al. 1984) and (Quinlan 1986).

The most common way to build decision trees is by top-down partitioning, starting with the full training set and recursively finding a univariate split that maximizes some local criterion (e.g. gain ratio) until the class distributions in the leaf partitions are sufficiently pure. The tree obtained by this process is usually too big and overfits the data, so it is pruned by examining each intermediate node and evaluating the utility of replacing it with a leaf. Pessimistic Error Pruning (Quinlan 1993) uses statistically motivated heuristics to determine this utility, while Reduced Error Pruning (Quinlan 1987) estimates it by testing the alternatives on a separate independent pruning set.

The class label assigned by each leaf node is determined by choosing the most frequent class label of the local training cases. After pruning, the local set of training cases for a leaf node can become quite large, and just taking the majority class might not capture enough of the structure still hidden in this set.

That is the starting point for our investigation to be described in the present paper. We will test the usefulness of a simple idea: we extend the basic decision tree learning algorithm so that instead of the majority rule, optionally a different kind of model can be used in any of the leaves. The decision of whether to replace a simple leaf by an alternative model is made during post-pruning. The resulting hybrid algorithms combine the (possibly very different) *biases* of top-down, entropy-based decision tree induction and the respective alternative leaf models. Specifically, we will test three simple algorithms, with rather different biases, as possible leaf models: a classifier based on linear regression, a simple nearest neighbor algorithm, and the well-known Naive Bayes classifier. We are interested in finding out whether this simple way of combining algorithms with different bias leads to more effective learners — in terms of predictive accuracy, or at least in terms of stability (i.e., reliable performance over a wider range of classification problems).

Learning Algorithms

The hybrid learning algorithms presented in this paper are based on the decision tree learning algorithm J48, a reimplementation of C4.5R8 (Quinlan 1993) within the *Waikato Environment for Knowledge Analysis (WEKA)*.[1] WEKA is a well-documented comprehensive implementation of many classification and regression learners, and allows the quick implementation of new or modified learning algorithms.

As mentioned above, we have implemented three hybrid tree learners, each based on J48, but with the possibility of using one of three alternative models in the leaves:

- **J48-Linear:** each leaf may contain a classifier that uses linear regression functions to approximate class membership (the so-called *ClassificationViaRegression* classifier in WEKA (Witten & Frank 1999)). That is, a linear regression function is learned for every class (trained with target values 1 for class members, 0 for all others), and for a new instance the class that gets the highest value is predicted. In the following, we will call this algorithm *Linear* (short for *Multi-response Linear Regression*).
- **J48-IB1:** a leaf may contain a simple nearest neighbor classifier (Cover & Hart 1967) using one neighbor (i.e., IB1, in the terminology of (Aha et al. 1991)).
- **J48-Bayes:** a leaf may contain a Naive Bayes Classifier (Langley et al. 1992) that uses a normal distribution as-

[1] The JAVA source code of WEKA has been made available at `www.cs.waikato.ac.nz`, see also (Witten & Frank 1999).

```
for each subtree (bottom-up) do
   train alternative model on local
    training set;
   evaluate original subtree on
    pruning set;
   evaluate alternative model on
    pruning set;
   evaluate local majority class
    prediction on pruning set;
   choose alternative with lowest error;
end
```

Figure 1: Schema of modified reduced error pruning. Alternative models are Naive Bayes for J48-Bayes, IB1 for J48-IB1 and Linear for J48-Linear.

sumption for the continuous attributes (John & Langley 1995).

In each hybrid algorithm, the unpruned tree is initially created exactly as in J48, using information gain ratio as the split criterion. The alternative leaf classifiers are then optionally introduced during the post-pruning phase, in which we use *Reduced Error Pruning (REP)*. In REP, the decision of replacing a subtree by a leaf node is based on a comparison of the error estimates of sub-tree and potential leaf obtained by using a separate pruning set. In the hybrid versions of the J48 algorithm, we allow any subtree to be replaced by either a leaf with majority class prediction (as in standard REP) *or* by a local learning algorithm that is trained on the local training examples. A subtree will be replaced if one of the two alternative models yields a lower error estimate than the respective subtree on the pruning set. This process is repeated until no subtree can be replaced anymore. A pseudo-code representation of our modified reduced error pruning procedure is shown in figure 1.

Empirical Evaluation

For empirical evaluation of the three hybrid learning algorithms we used twenty-seven datasets from the UC Irvine Machine Learning Repository (Blake & Merz 1998) which are listed in table 1. We evaluated each of the hybrid algorithms and, for comparison, the unmodified J48 algorithm (with standard Reduced Error Pruning — henceforth J48-R) and the algorithms that were used as alternative leaf models, i.e. Linear, IB1, and Naive Bayes. The four latter ones will be called *base learners* from now on.

Ten runs of ten-fold stratified cross validation each were used for evaluation. Average classification errors and standard deviations can be found in table 2 for the base learning algorithms J48, NaiveBayes, IB1 and Linear, and in table 3 for the hybrid learners. Table 3 also shows the final sizes of the trees generated on the complete training set.

A first look at the average performances over all datasets (see summary lines in tables 2 and 3) indicates a certain performance improvement for the three hybrid algorithms: the average error rates for the hybrid algorithms are lower than both J48's and the three alternative base learners' results.

In order to find out if the hybrid algorithms improve on

Dataset	Cl	Inst	D	C	dAcc	E
audiology	24	226	69	0	25.22	3.51
autos	7	205	10	16	32.68	2.29
balance-scale	3	625	0	4	45.76	1.32
breast-cancer	2	286	10	0	70.28	0.88
breast-w	2	699	0	9	65.52	0.93
colic	2	368	16	7	63.04	0.95
credit-a	2	690	9	6	55.51	0.99
credit-g	2	1000	13	7	70.00	0.88
diabetes	2	768	0	8	65.10	0.93
glass	7	214	0	9	35.51	2.19
heart-c	5	303	7	6	54.46	1.01
heart-h	5	294	7	6	63.95	0.96
heart-statlog	2	270	0	13	55.56	0.99
hepatitis	2	155	13	6	79.35	0.74
ionosphere	2	351	0	34	64.10	0.94
iris	3	150	0	4	33.33	1.58
labor	2	57	8	8	64.91	0.94
lymph	4	148	15	3	54.73	1.24
p.-tumor	22	339	17	0	24.78	3.68
segment	7	2310	0	19	14.29	2.81
sonar	2	208	0	60	53.37	1.00
soybean	19	683	35	0	13.47	3.84
vehicle	4	846	0	18	25.41	2.00
vote	2	435	16	0	61.38	0.96
vowel	11	990	3	10	9.09	3.46
waveform	3	5000	0	40	33.84	1.58
zoo	7	101	16	2	40.59	2.41

Table 1: The used datasets with number of classes and instances, discrete and continuous attributes, baseline accuracy (%) and a priori entropy in bits per instance (Kononenko & Bratko 1991).

J48 and the alternative base learners, we determined how often there is a significant difference of the error estimates. This was done using t-tests with a significance level of 95%. In table 3, significant differences are shown as + and − in columns RL, RI, and RB respectively. The first sign relates to J48 and the second to the corresponding alternative classifier. Insignificant error differences are shown as empty.

J48-Linear is significantly better than J48 on seventeen of the total twenty-seven datasets, and never significantly worse. J48-Bayes is better on seventeen datasets and worse on only one, J48-IB1 is better on twelve and worse on two datasets. Obviously, the multi-response linear regression algorithm is best suited to extend the original J48 algorithm.

In some cases, the original tree gets pruned back to a single leaf, effectively substituting the decision tree model with the learning algorithm used for the alternative leaf model. In these cases using the alternative algorithm in the first place would certainly be preferrable, since in the hybrid algorithm, part of the training set has to be reserved for pruning.

When we compare every hybrid algorithm directly with its alternative leaf classifier, the results are more moderate: J48-Linear is significantly better than Linear alone in nine cases and worse in ten (including three cases where the tree was pruned to a single leaf). J48-IB1 is better in nine cases

Dataset	J48-R	Linear	IB1	NaiveBayes
audiology	25.44±1.87	20.93±0.98	21.90±0.56	27.79±0.65
autos	29.61±2.40	34.59±1.77	25.95±1.00	42.24±1.26
balance-scale	21.22±1.25	13.38±0.58	13.25±0.55	9.50±0.29
breast-cancer	29.65±1.69	28.85±1.01	26.96±0.98	26.89±0.63
breast-w	5.41±0.74	4.21±0.14	4.78±0.23	3.95±0.12
colic	14.84±0.48	18.12±0.94	20.92±0.18	21.71±0.45
credit-a	14.75±0.46	14.46±0.27	18.84±0.71	22.16±0.17
credit-g	27.46±1.16	24.23±0.43	27.62±0.67	25.02±0.41
diabetes	26.42±1.23	23.03±0.46	29.40±0.53	24.27±0.28
glass	32.94±3.54	43.41±2.02	30.28±1.10	52.52±1.38
heart-c	23.27±2.54	15.41±0.75	24.09±0.70	16.17±0.35
heart-h	19.86±0.84	13.61±0.42	21.70±0.78	15.61±0.52
heart-stalog	21.81±1.74	16.22±0.89	24.04±0.95	15.63±0.62
hepatitis	19.61±1.98	16.26±1.35	18.97±1.30	16.13±0.96
ionosphere	10.66±1.34	13.45±0.48	13.19±0.47	17.41±0.43
iris	6.60±1.15	15.73±0.78	4.80±0.61	4.73±0.49
labor	20.18±4.47	12.46±1.74	14.74±2.22	6.14±1.70
lymph	24.46±2.79	15.07±1.72	18.38±1.68	16.76±0.77
primary-tumor	59.76±0.71	53.42±0.55	60.03±0.64	50.65±0.79
segment	4.51±0.41	16.77±0.09	2.89±0.13	19.90±0.18
sonar	28.56±3.07	27.55±1.09	13.51±0.66	32.02±1.27
soybean	11.95±0.73	6.31±0.13	8.96±0.22	7.10±0.21
vehicle	29.39±1.15	25.77±0.58	30.83±0.59	55.20±0.74
vote	4.64±0.61	4.37±0.00	7.36±0.29	9.82±0.16
vowel	27.40±1.35	57.08±0.68	0.89±0.15	37.19±1.00
waveform-5000	23.88±0.46	13.66±0.22	26.53±0.30	19.98±0.08
zoo	9.21±3.20	7.43±1.88	3.96±0.00	4.95 ± 0.00
Average	21.24	20.58	19.07	22.28

Table 2: Classification errors (%) and standard deviations for base learners: J48 with reduced error pruning, Linear, IB1 and NaiveBayes.

and worse in eight (including two trees of size one), and J48-Bayes is better in 13 cases and worse in 11 (three trees of size one). This seems to indicate that for datasets where the alternative algorithm would be the better choice, the hybrid tree is either not pruned enough, or the reduction of the available training data due to the necessity of a separate pruning set reduces the quality of the alternative models.

There are four cases where a hybrid algorithm is better than *all* base algorithms (i.e., both its constituent algorithms and also the other ones), namely J48-Bayes on ionosphere and J48-Linear on vehicle, iris and ionosphere. The accuracy difference between J48-Linear and J48-Bayes is insignificant on ionosphere. Expecting the hybrid learners to be consistently better than all of the base learners is clearly unrealistic.

If we view the hybrid algorithms as attempts at improving the underlying decision tree learner J48 by more flexibly adjusting its bias, the attempt can be considered a success (though maybe not a spectacular one); the hybrids produced significant improvement over J48 in 46 out of 81 cases, significant losses only in 3 cases. Of the three variants tested, J48-Linear and J48-Bayes seem to be preferable.

Of course, computing alternative leaf models for every node comes with a non-negligible computational cost. What we described in section "Learning Algorithms" is the most naive way of implementing the hybrid classifiers. We have a number of ideas on how to reduce the additional computation needed and are currently testing several alternatives.

Related Work

Combinations of decision tree with other learning algorithms have been studied in various ways before. An early example of a hybrid decision tree algorithm is presented in (Utgoff 1988). Here, a decision tree learner is introduced that uses linear threshold units at the leaf nodes; however, pruning is not considered as the algorithm was expected to work on noise-free domains.

In (Kohavi 1996) a decision tree learner named NBTree is introduced that has Naive Bayes classifiers as leaf nodes and uses a split criterion that is based directly on the performance of Naive Bayes classifiers in all first-level child nodes (evaluated by cross-validation) — an extremely expensive procedure. The tree size is determined by a simple stopping criterion and no postpruning is done. As in our experiments, the results reported in (Kohavi 1996) are mildly positive; in most cases, NBTree outperforms one, in a few cases both of its constituent base learners. In (Kohavi 1996) it is also pointed out that the size of the trees induced by

Dataset	J48-R	size	J48-Linear	size	R L	J48-IB1	size	R I	J48-Bayes	size	R B
audiology	25.44±1.87	42	23.45±1.57	37	+ −	23.23±1.26	31	+ −	24.12±1.80	36	+
autos	29.61±2.40	42	26.63±2.80	38	+ +	25.51±1.69	37	+	27.71±2.72	34	+
balance-scale	21.22±1.25	55	11.76±0.76	21	+ +	19.50±1.12	13	+ −	11.42±1.19	1	+ −
breast-cancer	29.65±1.69	22	30.31±2.19	19		30.03±1.85	16	−	29.55±1.77	19	−
breast-w	5.41±0.74	3	4.66±0.69	7	+	4.61±0.40	1	+	4.59±0.66	1	+ −
colic	14.84±0.48	64	15.57±1.42	56	+	16.20±0.81	58	− +	15.73±0.93	40	− +
credit-a	14.75±0.46	43	14.71±0.85	6		15.13±0.41	41	+	15.03±0.60	56	+
credit-g	27.46±1.16	64	27.26±1.19	40	−	28.10±1.02	73		26.89±0.84	47	−
diabetes	26.42±1.23	15	24.48±0.67	1	+ −	27.01±1.10	19	+	25.12±1.21	11	+ −
glass	32.94±3.54	27	33.27±3.49	5	+	31.78±2.86	21		32.71±3.15	23	+
heart-c	23.27±2.54	21	19.87±2.33	21	+ −	22.84±2.24	13		18.35±2.12	12	+ −
heart-h	19.86±0.84	8	17.59±1.52	3	+ −	20.78±1.07	20	− +	18.88±1.55	8	−
heart-statlog	21.81±1.74	25	19.37±1.34	1	+ −	21.85±1.17	21	+	18.96±2.39	1	+ −
hepatitis	19.61±1.98	1	17.74±2.22	15		20.00±2.60	1		17.61±2.02	1	+
ionosphere	10.66±1.34	9	9.23±1.30	5	+ +	9.46±1.37	5	+	9.46±0.96	5	+ +
iris	6.60±1.15	9	3.80±1.22	3	+ +	5.40±1.19	1	+	4.93±0.84	1	+
labor	20.18±4.47	7	20.53±4.53	3	−	16.67±5.50	5		8.60±4.09	1	+
lymph	24.46±2.79	18	19.66±2.58	13	+ −	20.41±2.42	16	+ −	19.80±2.23	9	+ −
p.-tumor	59.76±0.71	47	56.02±2.78	58	+ −	59.44±1.61	46		54.22±1.53	54	+ −
segment	4.51±0.41	59	4.32±0.52	47	+	3.16±0.28	25	+ −	4.05±0.50	51	+ +
sonar	28.56±3.07	7	27.60±2.30	1		17.69±2.84	1	+ −	25.34±2.09	13	+ +
soybean	11.95±0.73	120	6.68±0.79	28	+	8.83±0.38	31	+	7.77±0.87	37	+ −
vehicle	29.39±1.15	59	21.19±1.26	13	+ +	28.25±0.96	69	+ +	28.56±1.16	39	+
vote	4.64±0.61	9	4.64±0.53	9		4.90±0.74	9	+	4.46±0.63	11	+
vowel	27.40±1.35	183	16.51±1.14	73	+ +	3.84±0.55	1	+ −	21.82±0.90	121	+ +
waveform	23.88±0.46	187	14.18±0.18	1	+ −	23.69±0.54	175	+	17.63±0.55	17	+ +
zoo	9.21±3.20	13	6.44±1.70	9	+	5.15±1.02	3	+ −	3.76±0.78	5	+ +
Average	21.24	42.9	18.42	19.7		19.02	27.9		18.41	24.2	

Table 3: Classification errors (%) and standard deviations for J48 and the three hybrid variants. After each variant the tree size on the entire training set is shown. Plus/minus signs denote significantly better/worse classification error vs. J48-R and the appropriate base classifier - i.e. the hybrid's parents.

NBTree is often substantially smaller than the original C4.5 trees. The same can be observed in our experiments, but we do not attribute much practical significance to this fact. The hybrid trees may be smaller, but that does not necessarily make them more comprehensible to the user, due to the more complex models at the leaves.

In (Gama & Brazdil 1999) and (Gama 1999) a decision tree learner is described that computes new attributes as linear, quadratic or logistic discriminant functions of attributes at each node; these are then also passed down the tree. The leaf nodes are still basically majority classifiers, although the class probability distributions on the path from the root are taken into account. It is thus difficult to relate this method directly to our hybrid algorithms. However, introducing more complex tests at internal decision tree nodes can be interpreted as an alternative approach to modifying the bias of a decision tree learner, so a systematic comparison with our algorithms might be an interesting exercise.

A recursive bayesian classifier is introduced in (Langley 1993). The main idea is to split the data recursively into partitions where the conditional independence assumption holds. The experimental results reported in (Langley 1993) are somewhat disappointing, however; the author managed to show superiority of his method over simple Naive Bayes only on synthetic (and noise-free) data specifically generated for his experiments. Pre- and postpruning are also not implemented.

Model trees (Quinlan 1992) are similar to our hybrids in that they are decision trees with linear prediction models in the leaves; however, they predict numeric values rather than discrete classes (and thus they also use a different attribute selection criterion during tree construction).

Our hybrid learners are also related to general 'meta-learning' approaches like *stacking* (Wolpert 1992) or *cascade generalization* (Gama 1998), where different learning algorithms are combined by learning a classification model from the predictions of a set of base learners. There, the goal is to improve predictive accuracy by combining the opinions of several classifiers; the objective in our hybrid learners, on the other hand, is to derive specialized classifiers for different parts of the instance space. The relation between these two types of approaches might merit some more detailed investigation.

Conclusions and Further Work

To summarize, this paper has presented a first systematic study of three simple hybrid decision tree learning algorithms that can contain alternative models in their leaves. The experimental results to date indicate that some improvement over the original decision tree learner (and, in some cases, over both consituent algorithms or even over all base learners) is possible. In particular, the improvement over the decision tree learner J48 seems stable in the sense that the hybrids almost never perform significantly worse than J48.

- Substituting leaf nodes in the full tree. Currently the leaf nodes from the unpruned tree always use a majority model. Substitution with the alternative model only occurs if a subtree is replaced in the pruning process.
- Utilizing the whole training set for the leaf models (or at least including the pertinent examples from the pruning set). This could be done by re-training the model of each leaf node that already has replaced a subtree.
- Choosing among a set of more than two models to be used at a leaf node. Given a reasonable reduction of the computational cost, we will test a more general hybrid learner that is allowed to choose from a larger set of alternative classifiers for the leaves. That will take us closer to the goal of creating versatile learners that effectively construct classifiers with specialized bias optimized for different regions of the instance space.

Unfortunately, after implementing all these improvements we found that none improve the accuracy of our algorithm. Furthermore, detailed studies of the 10fold cross validations revealed that on most datasets, the choice of best leaf model (due to the last mentioned point) is highly variable even if there exists one best algorithm for this dataset. We presume this is due to the low number of examples that are present in a typical leaf. Further research is needed to compensate for this probable overfitting our algorithm seems to exhibit.

Acknowledgments

This research is supported by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung (FWF)* under grant P12645-INF, and by the ESPRIT long term research project METAL (project nr. 26357). The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Education, Science and Culture.

References

Aha, D.W.; Kibler, D.; and Albert, M.K. 1991. Instance-Based Learning Algorithms. *Machine Learning* 6(1).

Blake, C.L.; and Merz, C.J. 1998. UCI Repository of machine learning databases `http://www.ics.uci.edu/~mlearn/MLRepository.html`. Irvine, CA: University of California, Department of Information and Computer Science.

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

Cover, T.M.; and Hart, P.E. 1967. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory* IT-13(1).

Gama, J. 1998. Local Cascade Generalization. In Shavlik J.(ed.), *Proceedings of the 15th International Conference on Machine Learning (ICML '98)*, 206-214. Morgan Kaufmann, Los Altos/Palo Alto/San Francisco.

Gama, J. 1999. Discriminant Trees. In Bratko I. & Dzeroski S.(eds.), *Proceedings of the 16th International Conference on Machine Learning (ICML'99)* 134-142. Morgan Kaufmann, Los Altos/Palo Alto/San Francisco.

Gama, J.; and Brazdil, P. 1999. Linear Tree. *Intelligent Data Analysis* 3(1):1-22.

John, G.H.; and Langley, P. 1995. Estimating continuous distributions in Bayesian classifiers. *Proc. of 11th Conference on Uncertainty in Artificial Intelligence* 338-345. Montreal.

Kohavi, R. 1996. Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid. In Simoudis E. & Han J.(eds.), *KDD-96: Proceedings Second International Conference on Knowledge Discovery & Data Mining* 202-207. AAAI Press/MIT Press, Cambridge/Menlo Park.

Kononenko, I.; and Bratko, I. 1991. Information-Based Evaluation Criterion for Classifiers' Performance. *Machine Learning* 6(1).

Langley, P.; Iba, W.; and Thompson, K. 1992. An Analysis of Bayesian Classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence* 223-228. AAAI Press/MIT Press, Cambridge/Menlo Park.

Langley, P. 1993. Induction of Recursive Bayesian Classifiers. In Brazdil P.B.(ed.), *Machine Learning: ECML-93* 153-164. Springer, Berlin/Heidelberg/New York/Tokyo.

Quinlan, J.R. 1986. Induction of Decision Trees. *Machine Learning* 1(1):81-106.

Quinlan, J.R. 1987. Simplifying Decision Trees. *International Journal of Man-Machine Studies* 27:221-234.

Quinlan, J.R. 1992. Learning with Continuous Classes. *Proceedings of the Australian Joint Conference on Artificial Intelligence* 343-348. World Scientific, Singapore.

Quinlan, J.R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, Los Altos/Palo Alto/San Francisco.

Utgoff, P.E. 1988. Perceptron Trees: A Cast Study in Hybrid Concept Representations. In *Proceedings of the 7th National Conference on Artificial Intelligence* 601-605. Morgan Kaufmann, Los Altos/Palo Alto/San Francisco.

Witten, I.H.; and Frank, E. 1999. *Data Mining*. Morgan Kaufmann, Los Altos/Palo Alto/San Francisco.

Wolpert, D.H. 1992. Stacked Generalization. *Neural Networks* 5:241-249.