

An Empirical Comparison of Pruning methods for Ensemble Classifiers

Terry Windeatt, Gholamreza Ardeshir

Centre for Vision, Speech and Signal Processing
School of Electronics Engineering, Information Technology and Mathematics,
Guildford, Surrey, Gu2 7XH, UK
email: T.Windeatt, g.ardeshir@eim.surrey.ac.uk

Abstract. Many researchers have shown that ensemble methods such as Boosting and Bagging improve the accuracy of classification. Boosting and Bagging perform well with unstable learning algorithms such as neural networks or decision trees. Pruning decision tree classifiers is intended to make trees simpler and more comprehensible and avoid over-fitting. However it is known that pruning individual classifiers of an ensemble does not necessarily lead to improved generalisation. Examples of individual tree pruning methods are Minimum Error Pruning (MEP), Error-based Pruning (EBP), Reduced-Error Pruning (REP), Critical Value Pruning (CVP) and Cost-Complexity Pruning (CCP). In this paper, we report the results of applying Boosting and Bagging with these five pruning methods to eleven datasets.

1 Introduction

The idea of ensemble classifiers (also known as multiple classifiers) is based on the observation that achieving optimal performance in combination is not necessarily consistent with obtaining the best performance for a single classifier. However certain conditions need to be satisfied to realise the performance improvement, in particular that the constituent (base) classifiers be not too highly correlated [24]. Various techniques have been devised to reduce correlation between classifiers before combining including: (i) reducing dimension of training set to give different feature sets, (ii) incorporating different types of base classifier, (iii) designing base classifiers with different parameters for same type of classifier, (iv) resampling training set so each classifier is specialised on different subset, and (v) coding multi-class binary outputs to create complementary two-class problems. In this paper, we consider two popular examples belonging to category (iv), that rely on perturbing training sets.

Training on subsets appears to work well for unstable classifiers, such as neural networks and decision trees, in which a small perturbation in the training set may lead to a significant change in constructed classifier. Effective methods for improving unstable predictors based on perturbing the training set prior to combining, include Bagging [2] and Boosting [11]. Bagging (from Bootstrap Aggregating) forms replicate training sets by sampling with replacement, and

combines the resultant classifications with a simple majority vote. Boosting, which combines with a fixed weighted vote is more complex than Bagging in that the distribution of the training set is adaptively changed based upon the performance of sequentially constructed classifiers. Each new classifier is used to adaptively filter and re-weight the training set, so that the next classifier in the sequence has increased probability of selecting patterns that have been previously misclassified. In [3], the Bagging family is said to perform better with large tree base classifiers while the Boosting family is said to perform better with small trees. Several studies have compared Boosting and Bagging [1] [16] [9].

Size of tree can be changed by pruning, although as we explain in Sect. 2, the pruning process is not always simple. It might seem desirable to characterise performance as a function of degree of pruning. However pruning methods in general were not designed to operate with an independent parameter to change tree complexity in a smooth fashion and results are usually presented with or without pruning applied. There have been two recent large-scale studies of Bagging and Boosting with tree classifiers using MC4 [1] and C4.5 [9]. In [9], the test data was used to determine if pruning was required, on the assumption that internal cross-validation on the training set would come to the same conclusion. No pattern was discerned about whether to prune over thirty-three datasets, except that with Adaboost, pruning versus no-pruning did not register a difference on significance tests in any of the thirty-three domains. In [1], all experiments were carried out with pruning applied, but Bagging was repeated without pruning to help understand why Bagging produces larger trees. It appeared that trees generated from bootstrap samples were initially smaller than MC4, but ended up larger after pruning. The authors concluded that pruning reduces bias but increases variance. and that bootstrap replicates inhibit Reduced Error Pruning.

Ensemble classifiers have been empirically shown to give improved generalisation for a variety of classification data. However there is discussion as to why they appear to work well. Two concepts have been proposed for analysing behaviour, bias/variance from regression theory and the margin concept [12]. However they appear to provide useful perspectives for the way ensemble classifiers operate rather than an understanding of how ensemble classifiers reduce generalisation error. In absence of a complete theory, empirical results continue to provide useful insight to their operation.

In the past much effort has been directed toward developing effective tree pruning methods (for a review see [10]) in the context of a single tree. For a tree ensemble, besides individual tree (base classifier) pruning, it is also possible to consider ensemble pruning. In [7], five ensemble pruning methods used with Adaboost are proposed, but the emphasis there is on efficiency, i.e. finding a minimal number of base classifiers without significantly degrading performance. The goal of single tree pruning in an ensemble, is to produce a simpler tree base classifier which gives improved ensemble performance.

Decision trees can be divided into two categories according to type of test: multivariate decision trees and univariate trees. Multivariate decision trees are

those in which at each internal node several attributes are tested together while in univariate decision trees one attribute will be tested at each internal node. ID3 [17], C4.5 [18], CART [4] are examples of univariate decision tree and LMDT is a multivariate decision tree [22]. In this paper we have used C4.5, (which uses gain ratio in construction mode to determine which attribute to test), with several individual decision tree pruning methods.

In this paper, we briefly review five methods of decision tree pruning in section 2. These are Minimum Error Pruning (MEP), Error-based Pruning (EBP), Reduced-Error Pruning (REP), Critical Value Pruning (CVP) and Cost-Complexity Pruning (CCP). In section 3, we will explain the method used in the experiments and the results of Boosting and Bagging over eleven datasets when trees are pruned with different pruning algorithms.

2 Pruning Decision Trees

Decision tree pruning is a process in which one or more subtrees of a decision tree are removed. The need for pruning arises because the generated tree can be large and complex, so it may not be accurate or comprehensible. Complexity of a univariate decision tree is measured as the number of nodes, and the reasons for complexity are mismatch of representational biases and noise [5]. It means that the induction algorithm is unable to model some target concepts, and also that in some algorithms, (e.g. C4.5), subtree replication causes the tree to be too large and to overfit [17].

According to [10] there are different types of pruning methods but post-pruning is more usual [5]. The disadvantage of pre-pruning methods is that tree growth can be prematurely stopped, since the procedure estimates when to stop constructing the tree. A stopping criterion that estimates the performance gain expected from further tree expansion is applied, and tree expansion terminates when the expected gain is not accessible [10], [18]. A way around this problem is to use a post-pruning method which grows the full tree and retro-spectively prunes, starting at the leaves of the tree. Post-pruning methods remove one or more subtrees and replace them by a leaf or one branch of that subtree. One class of these algorithms divides the training set into a growing set and pruning set. The growing set is used to generate the tree as well as prune, while the pruning set is used to select the best tree [4]. In the case of shortage of training set, the cross-validation method is used i.e. the training set is divided into several equal-sized blocks and then on each iteration one block is used as pruning set and the remaining blocks used as a growing set. Another class of post-pruning algorithm uses all the training set for both growing and pruning [19]. However it is then necessary to define an estimate of the true error rate using the training set alone.

All the pruning methods considered here use post-pruning, and therefore construct the full tree before applying the pruning criteria.

2.1 Error-based pruning (EBP)

EBP was developed by Quinlan for use in C4.5. It does not need a separate pruning set, but uses an estimate of expected error rate. A set of examples covered by the leaf of a tree is considered to be a statistical sample from which it is possible to calculate confidence for the posterior probability of mis-classification. The assumption is made that the error in this sample follows a binomial distribution, from which the upper limit of confidence [15] is the solution for p of

$$CF = \sum_{x=0}^E \binom{N}{x} p^x (1-p)^{N-x} \quad (1)$$

where N is number of cases covered by a node and E is number of cases which is covered by that node erroneously (As C4.5 we have used an approximate solution for equation 1).

A default confidence level of 25% is suggested, and the upper limit of confidence is multiplied by the number of cases which are covered by a leaf to determine the number of predicted errors for that leaf. Further the number of predicted errors of a subtree is the sum of the predicted errors of its branches. If the number of predicted errors for a leaf is less than the number of predicted errors for the subtree in which that leaf is, then the subtree is replaced with the leaf. We tried changing the confidence level to vary the degree of pruning but found that this is not a reliable way of varying tree complexity [23].

2.2 Minimum Error Pruning (MEP)

MEP was introduced by Niblett and Bratko, and uses Laplace probability estimates to improve the performance of ID3 in noisy domains [14]. Cestnik and Bratko have changed this algorithm by using more general Bayesian approach to estimating probabilities which they called *m-probability estimation* [6]. In this algorithm, the parameter m is changed to vary degree of tree pruning. Their suggestion is that perhaps the parameter can be adjusted to match properties of learning domain such as noise. To prune a tree at a node, the first step is to calculate the expected error rates of its children. The expected error rate of a node is the minimum of $1 - p_i(t)$ where $p_i(t)$ is the probability of i th class of examples reaching that node and is determined by

$$p_i(t) = \frac{n_i(t) + p_{ai} \cdot m}{n(t) + m} \quad (2)$$

where $n_i(t)$ is the number of examples reaching the node and belong to the i th class, $n(t)$ is the total number of example reaching the node t and p_{ai} is the a priori probability of the i th class.

They called the expected error rate the *static error*. In the second step, *dynamic error* of the node is calculated, where *dynamic error* is defined as the weighted sum of the static errors of its children [6]. The node will be pruned if its static error is greater than its dynamic error and will be replaced by a leaf.

2.3 Reduced Error Pruning (REP)

REP requires a separate pruning set, and was proposed by Quinlan [19]. It simply replaces each internal node (non-leaf node) by the best possible branch with respect to error rate over the pruning set. Branch pruning is repeated until there is an increase in the pruning set error rate. The procedure is guaranteed to find the smallest, most accurate subtree with respect to the pruning set.

2.4 Critical Value Pruning (CVP)

CVP was proposed in [13], and operates with a variety of node selection measures. The idea is to set a threshold, the *critical value* which defines the level at which pruning takes place. An internal node is only pruned if the associated selection measures for the node and all its children do not exceed the *critical value*. The full tree is pruned for increasing critical values giving a sequence of trees, and then the best tree is selected on the basis of predictive ability. A number of suggestions were made for finding the best tree in the sequence, the obvious one being to use a separate pruning set as in REP.

2.5 Cost-complexity pruning (CCP)

CCP was developed for the CART system, and produces a sequence of trees by pruning those branches that give lowest increase in error rate per leaf over the training set [4]. Let the sequence of subtrees be denoted by $T_1 > T_2 > \dots > t$ in which T_1 is the original tree and T_2 has been generated by pruning T_1 and finally t is the root. Leaves of the subtree s are examined and assigned a measure α representing the increase in error rate per leaf

$$\alpha = \frac{M}{N(L(s) - 1)} \quad (3)$$

where N is the number of training examples, M is the additional number of misclassified examples when the leaf is removed, $L(s)$ is number of leaves in the subtree s [19]. To produce T_{i+1} from T_i , all nodes in T_i with the lowest α are pruned.

In order to select the best tree in the sequence, either cross-validation on the training set or a separate pruning set is employed. The selected tree is either

1. the smallest tree with error rate less than minimum observed error rate (0SE rule)
2. the smallest tree with error rate less than minimum observed error rate plus one standard error (1SE rule).

However in Breiman's Bagging [2], since each tree is built on a bootstrap replicate he uses the full training set to prune.

3 Experiments

To generate the decision tree that has been used as base classifier in the ensemble we have used C4.5 algorithm. After building the tree, we have used five different pruning algorithms MEP,EBP, REP, CVP, CCP where EBP is the pruning method used in the C4.5 system.

The datasets which have been used in the experiments can be found on UCI web site [21]. However, we have used the datasets which have been downloaded from Quinlan web site at the University of New South Wales [20], which have been split into training and test sets. Table 1 gives the description of the datasets, showing that, with the exception of Waveform-21, they have been divided into 70/30 training/testing split according to class distribution. All experiments have been carried out with number of base classifiers set to ten as in [16]. Each experiment is repeated ten times, and where a separate pruning set is employed this includes a random 70/30 growing/pruning split of the training set.

Table 1. Specification of Datasets

Name	Training set	Test set	Class	Attributes	
				Cont.	Disc.
Breast	466	233	2	10	–
BreastCancer	191	95	2	–	9
Crx	490	200	2	6	9
Glass	142	72	6	9	–
Heart	180	90	2	13	–
Hypothyroid	2108	1055	2	7	18
Iris	100	50	3	4	–
Labor-neg	40	17	2	8	8
Soybean-large	455	228	19	–	35
Vote	300	135	2	–	16
soybean-small	31	16	4	35	–
waveform-21	300	4700	3	21	–

In order to test whether degree of pruning could be varied to optimise generalisation performance we selected the MEP method, and varied parameter m as follows: $m = 0, .01, .5, 1, 2, 3, 4, 8, 12, 16, 32, 64, 128, 999, 9999$. For each value of m , resampling is started from the same random state. For values of $m > 4$ size of tree indicated that varying m was not a reliable way of varying degree of pruning, which agrees with the assessment in [10]. Table 2 shows for C4.5, Bagged C4.5 and Boosted C4.5 the ratio of unpruned error rate to minimum (over m) pruned error rate.

To have a comparison with cross-validation, we have applied 10-fold cross validation to the datasets, with results shown in table 4. We also applied Mc Nemar’s test with 5% confidence level [8] to determine whether difference between

Boosting and Bagging was significant, and table 4 shows the number of folds that showed a significant difference. Only two datasets appeared significantly different, Waveform21 and Soybean.

The results shown in table 3 are the average test errors of C4.5, Boosting and Bagging with EBP. The ratio of Boosting/Bagging error rates is shown for each dataset and averaged over all datasets. Similar tables were produced for the other pruning methods with average ratios: REP:0.98, CCP(0SE):1.03, CCP(1SE):1.03, CVP:1.02.

To compare the different pruning methods tables 5 and 6 show, for Boosting and Bagging respectively, the relative performance of the five pruning methods (for MEP $m = 2$). In tables 5 and 6 the pruning method with the minimum error rate is set to 1.00.

4 Discussion

Table 2 shows that, on average over these datasets, Boosting and Bagging can both benefit from varying parameter m in the MEP method. However, as noted previously as m increases for some datasets there was not a monotonic decrease with tree complexity. It should also be noted from table 4 that the average ratio for Boosting/Bagging for these datasets is 1.02, compared with a value of 0.93 over twenty-two datasets in [16].

The comparison of pruning methods in tables 5 and 6 shows that EBP performs best on average for Bagging and Boosting, and MEP worst. However the individual results indicate that there is no pattern corresponding to which pruning method performs best. For example, MEP performs best on three datasets. This suggests that if the type or level of pruning could be suitably chosen, performance would improve.

5 Conclusion

The fact that decision trees have a growing phase and pruning phase that are both data-dependent makes it difficult to match level or type of pruning to an ensemble of tree classifiers. Our results indicate that if a single pruning method needs to be selected then overall the popular EBP makes a good choice.

Acknowledgement

Gholamreza Ardeshtir is grateful to the Ministry of Culture and Higher Education of Iran for its financial support during his Ph.D studies.

References

1. E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1):105–142, 1999.
2. Leo Breiman. Bagging predictors. *Machine Learning*, 26(2):123–140, 1996.
3. Leo Breiman. Some infinity theory for predictor ensembles. Technical report, TR 577, Department of Statistics, University of California, Berkeley, 2000.

Table 2. Ratio of unpruned error rate to minimum pruned error rate using MEP

Name	C4.5	Boosted C4.5	Bagged C4.5
Breast	1.00	1.00	1.00
Breast-Cancer	1.72	1.07	1.23
Crx	1.18	1.00	1.00
Glass	1.04	1.09	1.00
Heart	1.00	1.15	1.06
Hypothyroid	1.00	1.00	1.00
Iris	1.00	1.00	1.00
Labor-neg	1.00	1.00	1.00
Soybean-large	1.01	1.01	1.00
Vote	1.75	1.25	1.25
Waveform-21	1.00	1.10	1.00
Average	1.15	1.06	1.05

Table 3. Mean Test Error of Boosting and Bagging and Ratio Boosting-Bagging using EBP

Name	C4.5	Boosted C4.5	Bagged C4.5	Ratio
Breast	5.58	3.86	3.86	1.00
Breast-Cancer	40.00	33.68	29.47	1.14
Crx	21.00	17.00	18.00	0.94
Glass	37.50	29.17	30.56	0.95
Heart	21.11	20.00	18.89	1.06
Hypothyroid	1.70	1.23	0.95	1.29
Iris	4.00	6.00	4.00	1.50
Labor-neg	29.41	23.53	23.53	1.00
Soybean-large	34.64	26.75	32.89	0.81
Vote	7.41	2.96	2.96	1.00
Waveform-21	30.74	20.09	22.09	0.91
Average				1.05

Table 4. Mean test error for 10-fold Cross Validation with Boosting and Bagging, and ratio Boosting-Bagging using EBP

Name	C4.5	Boosted C4.5	Bagged C4.5	Ratio	McNemar's Test
Breast	5.29	4.43	4.29	1.03	0
Breast Cancer	31.60	28.50	29.53	0.97	0
Crx	16.23	14.49	14.78	0.98	0
Glass	37.43	27.69	28.41	0.97	0
Heart	22.22	19.26	20.00	0.96	0
Hypothyroid	1.30	1.36	1.23	1.11	0
Iris	3.33	4.67	4.67	1.00	0
Labor-neg	41.67	28.33	31.67	0.89	0
Soybean-Large	42.42	35.27	24.71	1.43	2
vote	5.06	4.60	4.83	0.95	0
Waveform-21	25.14	17.32	18.40	0.94	1
Average				1.02	

Table 5. Relative Test Error of Boosting with different Puning Methods

Name	MEP	EBP	REP	CVP(0-SE)	CVP(1-SE)	CCP
breast	1.00	1.50	1.73	1.63	1.57	1.58
breastCancer	1.02	1.25	1.00	1.01	1.28	1.20
crx	1.02	1.02	1.02	1.00	1.07	1.06
glass	1.19	1.00	1.45	1.24	1.40	1.24
heart	1.00	1.38	1.35	1.37	1.45	1.26
hypothyroid	4.33	1.02	1.16	1.13	1.08	1.00
iris	1.07	1.07	1.00	1.04	1.04	1.00
labor-neg	1.00	1.00	1.02	1.12	1.18	1.18
soybean-large	2.11	1.16	1.02	1.00	1.05	1.00
vote	1.00	1.00	1.05	1.15	1.20	1.20
waveform-21	1.04	1.00	1.22	1.14	1.18	1.17
Average	1.50	1.13	1.18	1.17	1.23	1.17

Table 6. Relative Test Error of Bagging with different Puning Methods

Name	MEP	EBP	REP	CVP(0-SE)	CVP(1-SE)	CCP
breast	1.22	1.00	1.18	1.01	1.07	1.02
breastCancer	1.12	1.12	1.00	1.00	1.17	1.14
crx	1.00	1.06	1.07	1.00	1.03	1.03
glass	1.14	1.00	1.49	1.23	1.20	1.23
heart	1.00	1.00	1.09	1.09	1.08	1.11
hypothyroid	5.50	1.00	1.06	1.03	1.03	1.08
iris	1.00	1.00	1.50	1.40	1.60	1.50
labor-neg	1.00	1.00	1.20	1.25	1.43	1.02
soybean-large	2.73	1.30	1.15	1.00	1.02	1.04
vote	1.00	1.00	1.05	1.07	1.05	1.10
waveform-21	1.29	1.00	1.09	1.07	1.09	1.07
Average	1.64	1.04	1.17	1.11	1.16	1.12

4. Leo Breiman, J. H. Friedman, R. A. Olshen, and C.J. Stone. Classification and regression trees. *Wadsworth International Group*, 1984. <ftp://ftp.stat.berkeley.edu/pub/users/breiman>.
5. L. A. Breslow and D. W. Aha. Simplifying decision trees: A survey. *Knowledge Engineering Review*, pages 1–40, 1997.
6. B. Cestnik and I. Bratko. On estimating probabilities in tree pruning. In Kodratoff Y., editor, *Machine Learning - EWSL-91. European Working Session on Learning Proceedings*, pages 138–50. Springer-Verlag, 1991.
7. T. G. Dietterich D. Margineantu. Pruning adaptive boosting. In *International Conference on Machine Learning*, pages 211–218. Morgan Kaufmann, 1997.
8. Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.
9. T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–158, 2000.
10. F. Esposito, D. Malerba, and G. Semeraro. A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):476–491, May 1997.
11. Y. Freund and R. Schapire. Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156, 1996.
12. G. James. *Majority Vote Classifiers: Theory and Applications*. PhD thesis, Dept. of Statistics, Univ. of Stanford, May 1998. <http://www-stat.stanford.edu/gareth/>.
13. J. Mingers. Expert systems-rule induction with statistical data. *Operational Research Society*, 38:39–47, 1987.
14. T. Niblett and I. Bratko. Learning decision rules in noisy domains. In *Expert System 86, Cambridge*. Cambridge University Press, 1986.
15. J. Ross Quinlan. Personal communication from Quinlan.
16. J. Ross Quinlan. Bagging, boosting, and c4.5. In *Fourteenth National Conference on Artificial Intelligence*, 1996.
17. R. Quinlan. Induction of decision tree. *Machine Learning*, 1:81–106, 1986.
18. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California, 1993.
19. R.J. Quinlan. Simplifying decision trees. *International Journal of Man-Machine Studies*, 27:221–234, 1987.
20. <http://www.cse.unsw.edu.au/~quinlan/>.
21. These datasets can be found on: www.ics.uci.edu/mlearn/MLSummary.html.
22. P. E. Utgoff and C. E. Brodley. Linear machine decision trees. Technical report, Department of Computer Science, University of Massachusetts, Amhers, 1991.
23. T. Windeatt and G. Ardeshir. Boosting unpruned and pruned decision trees. In *Applied Informatics, Proceedings of the IASTED International Symposia*, pages 66–71, 2001.
24. T. Windeatt and R. Ghaderi. Binary labelling and decision level fusion. *Information fusion*, 2(2):103–112, 2001.