Simplifying Decision Trees by Pruning and Grafting: New Results
(Extended Abstract)

Floriana Esposito, Donato Malerba and Giovanni Semeraro
Dipartimento di Informatica - Università degli Studi di Bari - via Orabona 4, 70126 Bari, Italy
{esposito | malerbad | semeraro}@vm.csata.it

Abstract. This paper presents some empirical results on simplification methods of decision trees induced from data. We observe that those methods exploiting an independent pruning set do not perform uniformly better than the others. Furthermore, a clear definition of bias towards overpruning and underpruning is exploited in order to interpret empirical data concerning the size of the simplified trees.

1 Introduction

A major problem in top-down induction of decision trees (TDIDT) is the determination of the leaves [1]. One way to cope with it consists in keeping on growing a tree T_{max} in any case, and then retrospectively removing those branches that seem superfluous with respect to predictive accuracy. The final effect is that in this way the intelligibility of a decision tree is improved, without really affecting its predictive accuracy. Many methods have been proposed for simplifying decision trees; in [3] a review of some of them that employ *pruning* operators is presented. Informally, a pruning operator cuts a branch at a node t and removes the descendants of t itself. However, another complementary simplification operator, that we named *grafting*, has been employed in a well-known TDIDT system: C4.5 [9]. Briefly, a grafting operator substitutes a sub-branch of a node t onto the place of t iself, thus removing only some of the nodes of the subtree rooted in t. Simplification methods that use pruning and grafting operators are denoted with the general term of *pruning methods*.

In this paper we present the results of a wide experimentation on nine different pruning methods. In this empirical study, eleven databases taken from the UCI machine learning repository are considered. In order to detect possible biases of the methods towards underpruning or overpruning, we generate the smallest optimally pruned grown/trained tree for each experiment, and we compare the size of optimally pruned trees with the size of trees returned by the pruning method.

2 Experimental Design

The need of repeating experiments on some pruning methods arises from the fact that the experimental procedure designed by Mingers [6] to compare several pruning methods, in our opinion, presents some problems (see [3] for a detailed discussion).

In Table 1, the main characteristics of the data sets considered in our experiments are reported. Some of them have already been used to compare different pruning methods [6,8]. The database Heart is actually the join of four data sets on heart diseases, with the same number of attributes but collected in four distinct places (Hungary, Switzerland,

Table 1. Main characteristics of the databases used for the experimentation.

database	No. Cases	No. Classes	No. Attributes	Continuous attributes	Multi-valued attributes	Null values	% Base Error	Noise level	Uniform distrib.
Iris	150	3	4	4	0	no	66.67	low	yes
Glass	214	7	9	9	0	no	64.49	low	no
Led	1000	10	7	0	0	no	90	10%	yes
Hypo	3772	4	29	7	1	yes	7.7	no	no
P.-gene	106	2	57	0	57	no	50	no	yes
Hepat.	155	2	19	6	0	yes	20.65	no	no
Cleveland	303	2	14	5	5	yes	45.21	low	yes
Hungary	294	2	14	5	5	yes	36.05	low	no
Switzerland	123	2	14	5	5	yes	6.5	low	no
Long Beach	200	2	14	5	5	yes	25.5	low	no
Heart	920	2	14	5	5	yes	44.67	low	yes

Cleveland and Long-Beach). Only 14 out of the 76 original attributes have been selected, since they are the only ones deemed useful for the classification task. Moreover, examples have been assigned to two distinct classes: *no presence* (value 0 of the target attribute) and *presence* of heart diseases (values 1, 2, 3, 4).

In Table 1, columns headed "Continuous" and "Multi-valued" concern the number of attributes that are treated as real-valued and multi-valued discrete attributes respectively. All other attributes are binary. In the column "Null values", we simply report the presence of null values in at least one attribute of any observation. In fact, the system C4.5 used for building decision trees in our experiments provides us with a way of managing null values [9]. The column on base error refers to the percentage error obtained if the most frequent class is always predicted. We expect that good decision trees show a lower error rate than the base error. The last column states whether the distribution of examples per class is uniform or not.

In our experimental setup, each data set is randomly split into three subsets, according to the following criterion: *growing* set (49%), *pruning* set (21%) and *test* set (30%). The union of the growing and pruning set is called *training* set, and its size is just 70% of the whole data set. The growing set contains the 70% of cases of the training set, while the pruning set the remaining 30%. The growing set and the training set are used to learn two decision trees, which are called *grown* tree and *trained* tree respectively. The former is used by those methods that need an independent set in order to prune a decision tree, namely the reduced error pruning (REP) [8], the minimum error pruning (MEP) [2,7], the critical value pruning (CVP) [5], as well as those versions of the error complexity pruning based on a pruning set and adopting the 1SE rule (1SE) or not (0SE) [1]. Conversely, the trained tree is used by those methods that exploit the training set only, such as pessimistic error pruning (PEP) [8], error-based pruning (EBP) [9], as well as the cost-complexity pruning based on 10 cross-validation sets and adopting either the 0SE rule (CV-0SE) or the 1SE rule (CV-1SE). The evaluation of the error rate is always made on the test set.

For each data set employed, 25 trials are repeated by randomly partitioning the data set into three subsets. Moreover, for each trial two statistics are recorded: the number of leaves (*size*) of the resultant tree, and the error rate (*e.r.*) of the tree on the test set. This is done for pruned, grown and trained trees, so that a two-tailed paired t-test can be used to evaluate the significance of the error rate and size differences between trees.

As to MEP, the following m values have been chosen: 0.5, 1, 2, 3, 4, 8, 12, 16, 32, 64, 128, 512 and 1024. Experiments on the CVP are made by setting a maximum critical value equal to 1.0 and a step equal to 0.01. The only selection measure considered is the gain ratio [9].

3 Results and Conclusions

In order to study the effect of pruning on predictive accuracy of decision trees, we compare the error rates of the pruned trees with those of the corresponding trained trees. In practice, we compare two tree induction strategies: a *sophisticated* strategy that, in a way or another, prunes a large tree T_{max} constructed through recursive splitting, and a *naive* strategy that simply returns T_{max}. The main goal of this comparison is that of understanding whether tree simplification techniques are beneficial or not, at least for various databases considered in our experiments. Table 2 reports the results of the t-tests with a 0.1 confidence level. A (+) means that the application of the pruning method actually improves, on average, the predictive accuracy of the decision tree, while a (-) indicates a significant decrease in predictive accuracy. When the effect of pruning is neither good nor bad, a **0** is reported. It is easy to see that pruning does not generally decrease predictive accuracy. The only exception is represented by the application of the 1SE rule with cross-validation sets. Moreover, there is no clear indication that methods exploiting a pruning set perform definitely better than the others, as claimed in [5].

Another interesting characteristic of pruning methods is their tendency to overprune decision trees. In order to study such a problem, we produced two decision trees for each experiment, called *optimally pruned grown-tree* (OPGT) and *optimally pruned trained-tree* (OPTT) respectively. The former is a grown tree that has been pruned by using the reduced error pruning on the test set. Thus, it is the best pruned tree we could produce from the grown tree because of a property of optimality of the reduced error pruning [3]. Similarly, the OPTT is the best tree we could obtain by pruning some branches of the trained tree. Obviously, OPGTs are suitable to compare trees obtained with pruning methods that *do* use an independent pruning set, while OPTTs are more appropriate to compare results of pruning methods that *do not* need a pruning set. Therefore, by comparing the size of trees produced by a pruning method with the size of the corresponding optimal tree, we can have an indication of the tendency of each method. In Table 3, a summary

Table 2. Error rate variations for different pruning methods (significance level: 0.10)

database	REP	MEP	CVP	OSE	1SE	PEP	CV 0SE	CV 1SE	EBP
Iris	0	0	0	0	0	0	0	-	0
Glass	-	0	0	0	-	0	-	-	0
Led	-	-	-	0	-	0	0	-	0
Hypo	+	+	0	+	0	+	+	-	+
P. gene	0	0	0	0	0	0	0	0	0
Hepatitis	0	0	0	0	0	0	0	0	0
Cleveland	0	0	0	0	0	0	0	-	0
Hungary	+	+	0	+	+	+	+	+	+
Switzerland	+	0	+	+	+	+	+	+	+
Long Beach	+	+	+	+	+	+	+	+	+
Heart	0	0	0	0	0	+	0	-	+

Table 3. Tree size variations for different pruning methods (significance level: 0.10)

database	REP	MEP	CVP	OSE	1SE	PEP	CV 0SE	CV 1SE	EBP
Iris	-	-	u	-	o	-	-	o	u
Glass	-	u	u	-	o	u	-	o	u
Led	-	u	u	u	o	o	u	o	u
Hypo	o	u	u	-	o	-	-	o	u
P-gene	o	u	u	-	o	o	-	o	u
Hepatitis	-	u	-	-	o	-	o	o	u
Cleveland	-	-	u	-	o	u	o	o	u
Hungary	o	-	u	o	o	-	-	o	u
Switzerland	-	u	-	-	-	-	-	-	-
Long Beach	-	u	-	-	o	-	-	o	u
Heart	-	-	u	-	o	o	o	o	-

of the two-tailed paired t-tests at a significance level 0.1 is shown. Here, (**u**) stands for significant underpruning, (**o**) for significant overpruning, while (**-**) means no significant difference. At a glance, we can immediately conclude that MEP, CVP and EBP tend to underprune, while REP, 1SE and CV-1SE tend to overprune. We would be tempted to conclude that the predictive accuracy is improved whenever a pruning method does not produce trees with significant difference in size from the corresponding optimally pruned tree. However, this is not true for two reasons. First of all, it is not always true that an optimally pruned tree is more accurate than the corresponding grown/trained tree. In other words, pruning may help to simplify trees without improving its predictive accuracy. Secondly, tree size is a global feature that can provide us with an idea of what is happening, but it is not detailed enough to guarantee that only over or underpruning occurred. For instance, if a method overprunes a branch but underprunes another one, then it is actually increasing the error rate with respect to the optimal tree, but not necessarily the size. This problem can be observed with the database Glass and the method CV-0SE. Indeed, in this case there is a decrease in accuracy (see Table 2) but the size of pruned trees is close to the optimal value (see Table 3).

By ideally superimposing Tables 2 and 3 it is also possible to draw some other interesting conclusion. For instance, in some databases, such as Hungary and Heart, overpruning produces better trees than underpruning. This latter surprising result confirms Holte's observation that even simple rules perform well on most commonly used data sets in the machine learning community [4]. In any case, we have also indications that overpruning may have undesirable effects when too extremist, as in the case of the application of the rule 1SE.

References

1. Breiman, L., Friedman, J., Olshen, R., & C. Stone, *Classification and regression trees*, Belmont, CA: Wadsworth International, 1984.
2. Cestnik, B., & I. Bratko, On estimating probabilities in tree pruning. In Y. Kodratoff (Ed.), *Machine Learning - EWSL-91*, Lecture Notes in Artificial Intelligence, Berlin: Springer-Verlag, 138-150, 1991.
3. Esposito, F., Malerba, D., & G. Semeraro, Decision tree pruning as a search in the state space. In P. Brazdil (Ed.), *Machine Learning: ECML-93*, Lecture Notes in Artificial Intelligence, Berlin:Springer-Verlag, 165-184, 1993.
4. Holte, R.C., Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11, 63-90, 1993.
5. Mingers, J., Expert systems - rule induction with statistical data. *Journal of the Operational Research Society, 38*, 39-47, 1987.
6. Mingers, J., An empirical comparison of pruning methods for decision tree induction. *Machine Learning, 4*, 227 - 243, 1989.
7. Niblett, T., & I. Bratko, Learning decision rules in noisy domains. *Proceedings of Expert Systems 86*, Cambridge: Cambridge University Press, 1986.
8. Quinlan, J.R., Simplifying decision trees. *International Journal of Man-Machine Studies, 27*, 221-234, 1987.
9. Quinlan, J.R., *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann, 1993.