Comparing Simplification Methods for Model Trees with Regression and Splitting Nodes

Michelangelo Ceci, Annalisa Appice, and Donato Malerba

Dipartimento di Informatica, Università degli Studi
via Orabona, 4 - 70126 Bari - Italy
{ceci, appice, malerba}@di.uniba.it

Abstract. In this paper we tackle the problem of simplifying tree-based regression models, called model trees, which are characterized by two types of internal nodes, namely regression nodes and splitting nodes. We propose two methods which are based on two distinct simplification operators, namely pruning and grafting. Theoretical properties of the methods are reported and the effect of the simplification on several data sets is empirically investigated. Results are in favor of simplified trees in most cases.

1 Introduction

Model trees are tree-structured regression models that associate leaves with multiple linear regression functions. Internal nodes are typically spliting tests that partition the space spanned by m independent (or predictor) random variables x_i (both numerical and categorical). Regression models at the leaves capture the linear dependence between one or more independent variables and the continuous dependent (or response) variable y, locally to a partition of the sample space. Several methods have been proposed for the construction of the tree and for the estimation of the linear dependence at the leaves on the basis of a training sample. They have been implemented in some well-known model tree induction systems such as M5 [10], RETIS [5], M5' [14], HTL [12], TSIR [6] and SMOTI [7]. All these systems perform a *top-down* induction of model trees (TDIMT). However, the last two systems are characterized by two types of internal nodes: *regression nodes*, which perform only straight-line regressions, and *splitting nodes*, which partition the sample space. The regression model at a leaf is obtained by combining the straight-line regression functions associated to the regression nodes along the path from the root to the leaf. In SMOTI, the composition of straight-line regressions can be statistically interpreted as a multiple linear model built *stepwise*.

When building model trees, it is common practice to discard parts of the tree that describe spurious effects in the training sample rather than true features of the underlying phenomenon. The application of model tree simplification (pruning) methods follows the generation (growing) of the tree itself and tries avoid the overfitting problem under control. Several simplification methods have been reported in the literature, most of which are derived from those developed for decision trees [4]. For instance, RETIS bases its pruning algorithm on Niblett and Bratko's method

N. Zhong et al. (Eds.): ISMIS 2003, LNAI 2871, pp. 49–56, 2003.

[8], extended later by Cestnik & Bratko [2]. M5 uses a pessimistic-error-pruning-like strategy since it compares the error estimates obtained by pruning a node or not. The error estimates are based on the training cases and corrected in order to take into account the complexity of the model in the node. Similarly, in M5' the pruning procedure makes use of an estimate, at each node, of the expected error for the test data. The estimate is the resubstitution error compensated by a factor that takes into account the number of training examples and the number of parameters in the linear model associated to the node [14]. A method *à la* error-based-pruning is adopted in HTL, where the upper level of a confidence interval of the resubstitution error estimate is taken as the most pessimistic estimate of the error node [12]. A different solution has been proposed by Robnik-Sikonja and Kononenko [11] who applied the MDL principle. This principle is based on the coding of the possible solutions to the problem and the selection of the instance with the shortest code as the result.

A common characteristic of all these methods is that they have been defined for model trees whose internal nodes are only splitting tests. Since SMOTI has a different tree structure, it is necessary to develop new methods that correctly operates on the two types of internal nodes. It is noteworthy that no simplification method was proposed in TSIR, the only other system that induces trees with two types of nodes. In this paper, two methods are proposed: they are based on two distinct simplification operators, namely pruning and grafting. They are described in Section 3, after a brief introduction of SMOTI (next section). Experimental results are reported and discussed in Section 4.

2 Stepwise Induction of Model Trees

In this section we briefly recall some characteristics of SMOTI. A more detailed explanation of SMOTI and a comparison with other TDIMT methods are reported in [7]. In SMOTI the top-down induction of models trees is performed by considering *regression steps* and *splitting tests* at the same level. The former compute straight-line regression, while the latter partition the feature space. They pass down observations to their children in two different ways. For a splitting node t, only a subgroup of the $N(t)$ observations in t is passed to each child (left or right). No change is made on training cases. For a regression node t, all the observations are passed down to its only child, but the values of both the dependent and independent numeric variables not included in the multiple linear model associated to t are transformed in order to remove the linear effect of those variables already included. Thus, descendants of a regression node will operate on a modified training set. This is done in accordance to the statistical theory of linear regression, where the incremental construction of a multiple linear model is made by removing the linear effect of introduced variables each time a new independent variable is added to the model. In this way, a *multiple linear model* can be associated to each leaf. It involves all the numerical variables in the regression nodes along the path from the root to the leaf. Variables involved in regression nodes at top levels of the tree capture global effects, while those involved in regression nodes close to the leaves capture local effects.

During the tree growing phase, nodes are selected on the basis of an evaluation function. For a splitting node t it is defined as:

$$\sigma(t) = \frac{N(t_L)}{N(t)} R(t_L) + \frac{N(t_R)}{N(t)} R(t_R),$$

where $N(t)$ is the number of cases reaching the current splitting node t, $N(t_L)$ ($N(t_R)$) is the number of cases passed down to the left (right) child of t, and $R(t_L)$ ($R(t_R)$) is the resubstitution error of the left (right) child, computed as follows:

$$R(t_L) = \sqrt{\frac{1}{N(t_L)} \sum_{j=1}^{N(t_L)} (y_j - \hat{y}_j)^2} \quad \left(R(t_R) = \sqrt{\frac{1}{N(t_R)} \sum_{j=1}^{N(t_R)} (y_j - \hat{y}_j)^2} \right).$$

The estimate $\hat{y}_j = a_0 + \sum_{s=1}^{m} a_s x_s$ is computed by combining all univariate regression lines associated to regression nodes along the path from the root to t_L (t_R).

The evaluation of a regression step at node t cannot be naïvely based on the resubstitution error $R(t)$. Indeed, the splitting node "looks-ahead" to the best multiple linear regressions after the split is performed, while the regression step does not. A fairer comparison would be growing the model tree at a further level in order to base the computation of $\rho(t)$ on the best splitting node t' after the current regression node t is performed. Therefore, $\rho(t)$ is defined as follows:

$$\rho(t) = min \{R(t), \sigma(t')\}.$$

Both $\sigma(t)$ and $\rho(t)$ are compared to choose between three different possibilities: i) growing the model tree by adding a regression node t; *ii*) growing the model tree by adding a splitting node t; *iii*) stopping the tree's growth at node t.

Five different stopping criteria are used in SMOTI. The first performs the partial F-test to evaluate the contribution of a new independent variable to the model [3]. The second requires the number of cases in each node to be greater than a minimum value. The third operates when all continuous variables along the path from the root to the current node are used in regression steps and there are no discrete variables in the training set. The fourth creates a leaf if the error in the current node is below a fraction of the error in the root node [13, p. 60]. The fifth stops the growth when the *coefficient of determination* is greater than a minimum value [15, pp. 18-19].

3 Simplification Methods for Model Trees

The two proposed methods are both based on the use of an independent pruning set, but they adopt two distinct simplification operators (see Fig. 1), namely:

- the *pruning operator,* π_T, which associates each internal node t with the tree $\pi_T(t)$ having all the nodes of T except the descendants of t, and
- the *grafting operator*, γ_T, which associates each couple of internal nodes $<t,t'>$ directly connected by an edge with the tree γ_T ($<t,t'>$) having all nodes of T except those in the branch between t and t'.

The two methods are detailed in the next two subsections.

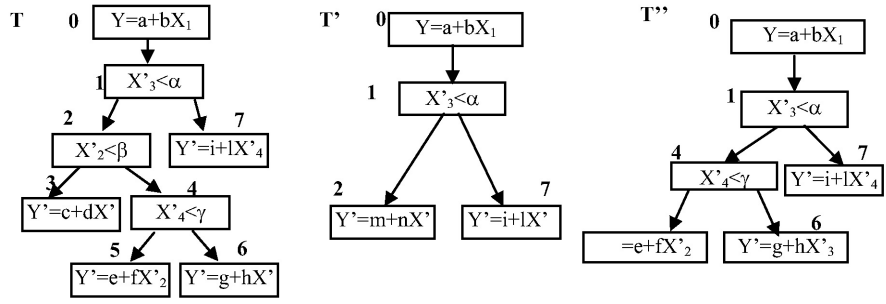

Fig. 1. The model tree T' is obtained by pruning T in node 2, while T" is obtained by grafting the subtree rooted in node 4 onto the place of node 2.

3.1 Reduced Error Pruning (REP)

This method is based on the Reduced Error Pruning (REP) proposed by Quinlan for decision trees [9]. It uses a pruning set to evaluate the goodness of the subtrees of a model tree T. The pruning set is independent of the set of observations used to build the tree T, therefore, the training set must be partitioned into a *growing* set used to build the tree and a *pruning* set used to simplify T.

The algorithm analyzes the complete tree T and, for each internal node *t*, it compares the mean square error (MSE) computed on the pruning set when the subtree T_t is kept, with the MSE computed on the same set when T_t is pruned and the best regression function is associated to the leaf *t*. The MSE is defined as follows:

$$MSE(T) = \frac{1}{N} \sum_{t \in \tilde{T}} \sqrt{\sum_{x_i \in t} (y_i - \hat{y}_i)^2} \ ,$$

where *N* is the number of examples $(\mathbf{x}_i, y_i)$ in the pruning set, $\tilde{T}$ is the set of leaves of the tree T, and $\hat{y}_i$ is the estimate of the response variable computed according to the multiple linear model associated to a leaf.

If the simplified tree has a better performance than the original one, it is advisable to prune T_t. The pruning is repeated on the simplified tree until further pruning increases the MSE. The nodes to be pruned are examined according to a bottom-up traversal strategy. When the node *t* is turned into a leaf the model associated to *t* is entirely determined on the basis of the growing set.

The following optimality theorem can be proven [1]:

Theorem. Given a model tree T constructed on a set of observations O and a pruning set O', the REP version that determines the regression model on O returns the smallest pruned subtree of T with the lowest error with respect to O'.

The specification "the REP version that determines the regression model on O" refers to the fact that once a node *t* has been pruned, the model associated to *t* is determined on the basis of the same growing set O. Alternatively, it could be determined on the basis of either the pruning set or the whole training set.

Finally, the computational complexity of REP is linear in the number of internal nodes, since each node is visited only once to evaluate the opportunity of pruning it.

3.2 Reduced Error Grafting

The Reduced Error Grafting (REG) is conceptually similar to REP and uses a pruning set to evaluate the goodness of T', a subtree of T. The algorithm operates recursively. It analyzes the complete tree T and, for each splitting node *t*, it compares the MSE made on the pruning set when the subtree T_t is kept, with the MSE made on the pruning set when T_t is turned into REG(T_{t_1}) or REG(T_{t_2}), where t_1 and t_2 are children of *t*. Sometimes, the simplified tree has a better performance than the original one. In this case, it appears convenient to replace *t* with its best simplified subtree (left of right). This grafting operation is repeated on the simplified tree until the MSE increases. The nodes to be pruned are examined according to a bottom-up traversal strategy.

This method is theoretically advantaged with respect to REP, since it allows the replacement of a subtree by one of its branches. Indeed, if *t* is a node that should be pruned according to some criterion, while *t'* is a child of *t* that should not be pruned according the same criterion, such simplification strategy either prunes and loses the accurate branch $T_{t'}$ or does not prune at all and keeps the inaccurate branch T_t. On the contrary, REG acts by grafting $T_{t'}$ onto the place of *t*, so saving the good sub-branch and deleting the useless node *t*.

Similarly to REP, a theorem on the optimality of the pruned tree can be proven [1].
Theorem. Given a model tree T constructed on a set of observations O and a pruning set O', the REG version that determines the regression model on O returns the smallest grafted subtree of T with the lowest error with respect to O'.

The complexity of REG is $O(|N_T| \log_2 |N_T|)$, where N_T is the set of nodes in T.

4 Comments on Experimental Results

The experiment aims at investigating the effect of simplification methods on the predictive accuracy of the model trees. REP and REG were implemented as a module of KDB2000 (http://www.di.uniba.it/~malerba/software/kdb2000/) and have been empirically evaluated on ten datasets listed in table 1.

Table 1. Datasets used in the empirical evaluation of SMOTI.

Dataset	No. Cases	No. Attributes	Continuous	Discrete
Abalone	4177	10	9	1
Auto-Mpg	392	8	5	3
Auto-Price	159	27	17	10
Bank8FM	4499	9	9	0
Cleveland	297	14	7	7
Housing	506	14	14	0
Machine CPU	209	10	8	2
Pyrimidines	74	28	28	0
Triazines	74	61	61	0
Wisconsin Cancer	186	33	33	0

Each dataset is analyzed by means of a 10-fold cross-validation. For every trial, the training set is partitioned into growing (70%) and pruning set (30%). SMOTI is trained on the growing set, pruned on the pruning set and tested on the hold-out block (testing set). Comparison is based on the average mean square error (*Avg.MSE*) made on the testing sets and on the average number of leaves. The stopping criteria used in the experimentation are fixed as follows: the significance level α used in the F-test is set to 0.075, the minimum number of cases falling in each internal node must be greater than the square root of the number of cases in the entire training set, the error in each internal node must be greater than the 0.01% of the error in the root node, the coefficient of determination in each internal node must be below 0.90 for model trees induced on the entire training set and 0.99 for model trees induced on the growing set and after simplified by means of REP or REG method.

Experimental results are listed in Table 2, which reports the average MSE of (un-pruned/pruned) SMOTI trees built on training/growing set. For comparison purposes, results obtained by M5' are reported as well. They show that pruning is generally beneficial since REP and REG decrease the average MSE of SMOTI trees built on the growing set. The two methods also drastically reduce the size of the induced trees, often of an order of magnitude, although REG tends to be more conservative than REP. The pruning method implemented in M5' outperforms both REP and REG in most data sets. However, the worst performance of REP and REG can be justified if we consider that M5' pruned a model tree which was originally more accurate than that pruned by REP and REG because of the full use of the cases in the training set.

This result is similar to that reported in [4] for decision trees. Even in that case, it was observed that methods requiring an independent pruning set are at a disadvantage. This is due to the fact that the set of pre-classified cases is limited and, if part of the set is put aside for pruning, it cannot be used to grow a more accurate tree. A clear example is represented by the Auto-Price dataset, where the average number of leaves of REG and M5' is the same (1.6) while the accuracy is different.

Table 2. Experimental results of pruning methods for SMOTI and M5'.

	SMOTI un-pruned trees				*SMOTI pruned trees*				*M5'*			
	Tree built on training set		*Tree built on growing set*		*REP*		*REG*		*Tree built on training set*		*Pruning*	
	Avg MSE	*Avg Leaves*	*Avg MSE*	*Avg Leaves*	*Avg MSE*	*Avg Leaves*	*Avg MSE*	*Avg Leaves*	*Avg MSE*	*Avg No Leaves*	*Avg MSE*	*Avg Leaves*
Abalone	2.5364	143	6.724	95.6	2.185	5.4	2.179	25.4	2.7724	281.4	2.126	11
AutoMpg	3.1493	13.7	4.4866	19.2	3.5633	3.1	3.7436	8.5	3.2010	22.6	2.835	4.6
Auto Price	2246.0	4.3	2481.7	8	2746.3	1.6	2890.4	4.1	2358.8	12.4	2390.1	1.6
Bank8FM	0.0383	2.2	0.0427	68.8	0.035	5.6	0.034	30.2	0.0409	417.7	0.0319	27
Cleveland	1.3160	21.7	1.521	17.3	0.914	2.3	0.934	5.2	1.2496	28.1	0.9028	1.6
Housing	3.58	8.8	5.717	19.6	4.080	3.1	3.912	7.6	4.2792	50.7	3.815	14.5
MachineCPU	55.314	4.0	71.699	6	70.953	2.7	69.145	2.4	57.352	12	58.341	3.8
Pyrimidines	0.1056	3.8	0.1872	6.4	0.1034	1.8	0.1352	1.8	0.0927	3.4	0.0864	3
Triazines	0.2017	16.6	0.1820	13.3	0.155	1.2	0.229	3.8	0.1550	9.1	0.131	3.5
Wisconsin	51.413	18.4	72.376	11.5	33.464	1.2	37.455	1.9	45.406	32.1	34.397	2.7

Table 3. Average percentage of the MSE for pruned trees w.r.t. the MSE of un-pruned trees. MSE is computed on the testing set. Best values are in bold.

Data Set	REP/unpruned SMOTI	REG/unpruned SMOTI	Pruned M5'/unpruned M5'
Abalone	32.49%	**32.40%**	76.68%
AutoMpg	**79.42%**	83.4%	88.56%
Auto Price	110.66%	116.46%	**101.32%**
Bank8FM	81.96%	79.62%	**77.99%**
Cleveland	**60.09%**	61.40%	72.24%
Housing	71.36%	**68.42%**	89.15%
Machine CPU	98.96%	**96.44%**	101.72%
Pyrimidines	**55.23%**	72.23%	93.20%
Triazines	85.71%	125.86%	**84.51%**
Wisconsin Cancer	**46.23%**	51.75%	75.75%

A different view of results is offered in Table 3, which reports a percentage of the Avg. MSE made by pruned trees on the testing sets with respect to the average mean square error made by un-pruned trees on the same testing sets. The table emphasizes the gain of the use of pruning. In particular, pruning is beneficial when the value is less than 100%, while it is not when the value is grather than 100%. Results reported confirm that pruning is beneficial for nine out of ten datasets. Moreover, the absolute difference of Avg. MSE for REP and REG is below 5% in seven datasets. Finally, it is worthwhile to notice that the gain of REP and REG is better than the corresponding gain of M5' pruning method in six datasets. This induces to hypothesize that the better absolute performances of M5' are mainly due to the fact that the tree to be pruned is more accurate because of the full use of training cases.

5 Conclusions

SMOTI is a TDIMT method which integrates the partitioning phase and the labeling phase. Similar to many decision tree induction algorithms, SMOTI may generate model trees that overfit training data. In this paper, the *a posteriori* simplification (or pruning) of model trees has been investigated in order to solve this problem. Two methods, named REP and REG, have been defined. They are both based on the use on an independent pruning set, but adopt different simplification operators. Some experimental results have been reported on the pruning methods and show that pruning is generally beneficial. Moreover, the comparison with another well-known TDIMT method, namely M5', which uses the training data both for growing and for pruning the tree, has shown that putting aside some data for pruning can lead to worse results. As future work, we plan to extend this comparison to other TDIMT systems (e.g. HTL and RETIS). Moreover, we intend to implement a new simplification method based on both pruning and grafting operators and to eventually extend MDL-based pruning strategies developed for regression [11] trees to the case of SMOTI trees. This extension should overcome problems we observed for small datasets since the new pruning algorithm will not require an independent pruning set.

Acknowledgments. The work presented in this paper is partial fulfillment of the research objective set by the MIUR COFIN-2001 project on "Method for the extraction, validation and representation of statistical information in decision context".

References

1. Ceci, M., Appice, A. & Malerba D.: Simplification Methods for Model Trees with Regression and Splitting Nodes. In P. Perner, & A. Rosenfeld (Eds.), *Machine Learning and Data Mining in Pattern Recognition,* Lectures Notes in Artificial Intelligence, 2734, Springer, Berlin (2003), in press.
2. Cestnik B. and Bratko I.: On estimating probabilities in tree pruning, *Proc. of the Fifth European Working Session on Learning*, Springer, (1991), 151–163.
3. Draper N.R. and Smith H.: *Applied regression analysis*, John Wiley & Sons, (1982).
4. Esposito F., Malerba D., Semeraro G.: A comparative analysis of methods for pruning decision trees. *IEEE Trans. PAMI*, Vol. 19, Num. 5, (1997), 476–491.
5. Karalic A.: Linear regression in regression tree leaves, in *Proceedings of ISSEK '92 (International School for Synthesis of Expert Knowledge)*, Bled, Slovenia, (1992).
6. Lubinsky D.: Tree Structured Interpretable Regression, in Learning from Data, Fisher D. & Lenz H.J. (Eds.), Lecture Notes in Statistics, 112, Springer, (1996), 387–398.
7. Malerba D., Appice A., Ceci M., Monopoli M. : Trading-off Local versus Global Effects of Regression Nodes in Model Trees. In H.-S. Hacid, Z.W. Ras, D.A. Zighed, Y. Kodratoff (Eds), *Proceedings of the 13th International Symposium, ISMIS'2002*, Lecture Notes in Artificial Intelligence, 2366, Springer, Berlino, Germania, (2002), 393–402.
8. Niblett, T. & Bratko I.: Learning decision rules in noisy domains. In Bramer, M. A., *Research and Development in Expert Systems III*, Cambridge University Press, Cambridge, (1986), 25–34.
9. Quinlan J.R.: Simplifying decision trees. *International Journal of Man-Machine Studies*; 27, (1987), 221–234.
10. Quinlan J.R.: Learning with continuous classes, in *Proceedings AI'92*, Adams & Sterling (Eds.), World Scientific, (1992), 343–348.
11. Robnik-Šikonja M., Kononenko I.: Pruning Regression Trees with MDL. In H. Prade (Ed.), Proceedings of the 13th European Conference on Artificial Intelligence, John Wiley & Sons, Chichester, England, (1998), 455–459.
12. Torgo L.: Functional Models for Regression Tree Leaves, in *Proceedings of the Fourteenth International Conference (ICML '97)*, D. Fisher (Ed.), Nashville, Tennessee, (1997).
13. Torgo L.: *Inductive Learning of Tree-based Regression Models*, Ph.D. Thesis, Department of Computer Science, Faculty of Sciences, University of Porto. (1999).
14. Wang Y. & Witten I.H.: Inducing Model Trees for Continuous Classes, in *Poster Papers of the 9th European Conference on Machine Learning (ECML 97)*, M. van Someren, & G. Widmer (Eds.), Prague, Czech Republic, (1997), 128–137.
15. Weisberg S.: *Applied regression analysis*, 2nd edn. New York: Wiley, (1985).