Trading-Off Local versus Global Effects of Regression Nodes in Model Trees

Donato Malerba, Annalisa Appice, Michelangelo Ceci, and Marianna Monopoli

Dipartimento di Informatica, Università degli Studi di Bari
via Orabona 4, 70125 Bari, Italy
{malerba, appice, ceci, monopoli}@di.uniba.it

Abstract. Model trees are an extension of regression trees that associate leaves with multiple regression models. In this paper a method for the top-down induction of model trees is presented, namely the Stepwise Model Tree Induction (SMOTI) method. Its main characteristic is the induction of trees with two types of nodes: regression nodes, which perform only straight-line regression, and split nodes, which partition the sample space. The multiple linear model associated to each leaf is then obtained by combining straight-line regressions reported along the path from the root to the leaf. In this way, internal regression nodes contribute to the definition of multiple models and have a "global" effect, while straight-line regressions at leaves have only "local" effects. This peculiarity of SMOTI has been evaluated in an empirical study involving both real and artificial data.

1 Introduction

Regression trees are well-known tree-based prediction models for numerical variables [1]. As in the case of decision trees, they are generally built top-down by recursively partitioning a feature space $\mathcal{X}$ spanned by m independent (or predictor) variables x_i (both numerical and categorical). The main difference is that the dependent (or response) variable y to be predicted is continuous. Therefore, each leaf in the tree is associated with a numerical value, and the underlying model function $y=g(\mathbf{x})$ is approximated by means of a piecewise *constant* one. *Model trees* generalize the concept of regression trees in the sense that they approximate the function above by a piecewise *linear* function, that is they associate leaves with multiple models. The problem of inducing model trees from a training set has received attention both in statistics [2,7,10] and in machine learning. Some of the model tree induction systems developed are: M5 [9], RETIS [4], M5' [13], TSIR [5], and HTL [11,12]. In most of them the multiple model associated with a leaf is built on the basis of those training cases falling in the corresponding partition of the feature space. Therefore, models in the leaves have only a "local" validity and do not consider the "global" effects that some variables might have in the underlying model function. Such global effects can be represented by variables that are introduced in the multiple models at higher levels of the model trees. However, this requires a different tree-structure, where internal nodes can either define a further partitioning of the feature space or introduce some regression variables in the models to be associated to the leaves.

M.-S. Hacid et al. (Eds.): ISMIS 2002, LNAI 2366, pp. 393-402, 2002.

In this paper we present the current state of the art of the research on top-down induction of model trees and we motivate the stepwise construction of models associated with the leaves. A new method, named Stepwise Model Tree Induction (SMOTI), is presented. SMOTI is characterized by the construction of tree models with both regression and split nodes. Regression nodes perform straight-line regression, while split nodes partition the sample space. The multiple linear model associated with each leaf is obtained by composing the effect of regression nodes along the path from the root to the leaf. Therefore, variables of the regression nodes selected at higher levels in the tree have a "global" effect, since they affect several multiple models associated with the leaves.

The state of the art of model tree induction is described in the next section, while in Section 3 the method SMOTI is introduced, and its computational complexity is analyzed. Finally, in Section 4 some experimental results are reported and the trade-off between "local" and "global" effects is discussed.

2 Induction of Model Trees: State of the Art

The induction of model trees can be reformulated as a search problem in the space of all possible model trees that can be built with m independent variables. Since an exhaustive exploration of this space is not possible in practice, several heuristics (*evaluation functions*) have been proposed to solve this problem. In CART [1], the quality of the (partially) constructed tree T is assessed by means of the mean square error $R^*(T)$, whose sample estimate is:

$$R(T) = \frac{1}{N} \sum_{t \in \tilde{T}} \sum_{x_i \in t} (y_i - \bar{y}(t))^2$$

where N is the number of training examples ($\mathbf{x}_i$, y_i), $\tilde{T}$ is the set of leaves of the tree, and $\bar{y}(t)$ is the sample mean of the response variable, computed on the observations in the node t. By denoting with $R(t)$ and $s^2(t)$ the resubstitution estimate of risk and the sample variance at a node t, respectively, $R(T)$ can be rewritten as:

$$R(T) = \sum_{t \in \tilde{T}} R(t) = \sum_{t \in \tilde{T}} \frac{N(t)}{N} s^2(t) = \sum_{t \in \tilde{T}} p(t) s^2(t)$$

where $N(t)$ is the number of observations in the node t and $p(t)$ is the probability that a training case reaches the leaf t. When the observations in a leaf t are partitioned into two groups, we obtain a new tree T', where t is an internal node with two children, say, t_L and t_R. Different splits generate distinct trees T', and the choice of the best split is made by minimizing the corresponding $R(T')$, that is, by minimizing $p(t_L)s^2(t_L)+ p(t_R)s^2(t_R)$, the contribution to $R(T')$ given by the split.

This heuristic criterion, initially conceived for regression trees, has also been used for model trees. In the system HTL the evaluation function is the same as that reported above, while in M5 the sample variance $s^2(t)$ is substituted by the sample standard deviation $s(t)$. The problem with these evaluation functions, when used in model tree induction, is that they do not take into account the models associated with the leaves of the tree. In principle, the optimal split should be chosen depending on how well each model fits the data. In practice, many model tree induction systems choose the optimal split on the basis of the spread of observations with respect to the *sample mean*. However, a model associated with a leaf is generally more sophisticated than the sample mean. Therefore, *the evaluation function is incoherent with respect to the model tree being built*. Consequently, the induced tree may fail to discover the underlying model, as exemplified in [6]. This problem is due to the neat separation of the *splitting* stage from the *predictive* one. The partitioning of the feature space (splitting stage) does not take into account the multiple regression models that can be associated with the leaves. Moreover, the association of models with the leaves (prediction stage) takes place only when the partition of the feature-space has been fully defined; therefore, it is difficult to establish whether a variable has a more global effect involving several regions of the feature space.

This problem does not occur in regression tree induction, since the models are the sample means which are used in the computation of *R(T)*. Moreover, the choice of a constant function (the sample mean) as the type of model in the leaves explicitly prevents the differentiation between global and local effects of variables in the models. For different reasons, the same problem cannot potentially occur in RETIS, whose heuristic criterion is to minimize $p(t_L)s^2(t_L)+ p(t_R)s^2(t_R)$, where $s^2(t_L)$ ($s^2(t_R)$) is now computed as the mean square error with respect to the regression plane g_L (g_R) found for the left (right) child:

$$s^2(t_L)=\frac{1}{N(t_L)}\sum_{x_i\in t_L}(y_i-g_L(x_i))^2 \qquad \left(s^2(t_R)=\frac{1}{N(t_R)}\sum_{x_i\in t_R}(y_i-g_R(x_i))^2\right)$$

In practice, for each possible partitioning the best regression planes at leaves are chosen, so that the selection of the optimal partitioning can be based on the result of the prediction stage.

The weakness of the RETIS heuristic evaluation function is its high computational complexity, especially when all independent variables are continuous. In particular, it can be proven that the choice of the first split takes time $O(N(N-1)m(m+1)^2)$, which is cubic in m and square in N [6]. In addition to the high computational cost, RETIS is characterized by models that can take into account only local decisions.

A solution to both problems is the stepwise construction of multiple linear models by intermixing regression steps with partitioning steps, as done in TSIR. TSIR has two types of node: split nodes and regression nodes. The former perform a boolean test on a variable and have two children. The latter compute a single variable regression, Y = a+bX, and pass down to its *unique* child the residuals y_i - (a+bx_i) as new values of the response variable. Thus, descendants of a regression node will operate on a modified training set. Lubinsky claims that “each leaf of the TSIR tree corresponds to a different multiple linear regression,” and that “each regression step adds one variable and its coefficients to an incrementally growing model” [5].

However, this interpretation is not correct from a statistical point of view, since the incremental construction of a multiple linear regression model is made *by removing the linear effect of the introduced variables each time a new independent variable is added to the model* [3]. For instance, let us consider the problem of building a regression model $Y=a+bX_1+cX_2$ through a sequence of straight-line regressions. We start regressing Y on X_1, so that the model $Y = a_1+b_1X_1$ is built. This fitted equation does not predict Y exactly. By adding the new variable X_2, the prediction might improve. Instead of starting from scratch and building a model with both X_1 and X_2, we can build a linear model for X_2 given X_1:

$X_2 = a_2+b_2X_1$,
then compute the residuals on X_2:
$X'_2 = X_2 - (a_2+b_2X_1)$,
and finally regress Y on X'_2 alone:
$Y = a_3 + b_3X'_2$.
By substituting the equation of X'_2 in the last equation we have:
$Y = a_3 + b_3X_2 -a_2b_3-b_2b_3X_1$.

It can be proven that this last model coincides with the first model built, that is $a=a_3-a_2b_3$, $b=-b_2b_3$ and $c=b_3$. Therefore, when the first regression line of Y on X_1 is built we do not pass down *the residuals of Y* but *the residuals of the regression of* X_2 *on* X_1. This means we remove the linear effect of the variables already included in the model (X_1) from those variables to be selected for the next regression step (X_2). TSIR operates in a different way, so that it is not possible to assert that the composition of straight-line models found along a path from the root to a leaf is equivalent to a multiple linear model associated with the leaf itself. In fact, the only correct interpretation is that the subtree of a regression node is in turn a model tree that aims at predicting the residuals of the regression performed in the node.

The above problem does not occur in the system SMOTI, which removes the effect of the variable selected by a regression node before passing down training cases to deeper levels. However, this adjustment must be accompanied by a look-ahead strategy when regression nodes and split nodes are compared for selection. This has been also taken into account in the design of SMOTI, as explained in the next section.

3 Induction of Model Trees in SMOTI

In SMOTI, the development of a tree structure is not only determined by a recursive partitioning procedure, but also by some intermediate prediction functions.

This means that there are two types of node in the tree: regression nodes and split nodes. The former performs only straight-line regressions, while the latter partitions the feature space. They pass down observations to their children in two different ways. For a split node t, only a subgroup of the $N(t)$ observations in t is passed to each child, and no change is made on the variables. For a regression node t, all the observations are passed down to its only child, but the values of the independent variables not included in the model are transformed, to remove the linear effect of those variables already included. Thus, descendants of a regression node will operate on a modified training set.

The validity of either a regression step on a variable X_i or a splitting test on the same variable is based on two distinct evaluation measures, $\pi(X_i,Y)$ and $\sigma(X_i,Y)$ respectively. The variable X_i is of a continuous type in the former case, and of any type in the latter case. Both $\pi(X_i,Y)$ and $\sigma(X_i,Y)$ are mean square errors,[1] therefore they

[1] This is different from TSIR, which minimizes the absolute deviation between a median (the model) and the Y values of the cases. Actually, the minimization of absolute deviation is more robust with respect to the presence of outliers and skewed distributions. However, SMOTI coherently minimizes the least squares both when a straight-line regression has to be built and when two different alternatives have to be compared.

can be actually compared to choose between either growing the model tree by adding a regression/split node *t*, or stopping the tree's growth at node *t*.

As pointed out in Section 2, the evaluation measure $\sigma(X_i,Y)$ should be coherently defined on the basis of the multiple linear model to be associated with each leaf. In the case of SMOTI it is sufficient to consider a straight-line regression associated with each leaf t_R (t_L), since regression nodes along the path from the root to t_R (t_L) already define partially a multiple regression model (see Figure 1a-b).

If X_i is continuous and α is a threshold value for X_i then $\sigma(X_i,Y)$ is defined as:

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

where $N(t)$ is the number of cases reaching *t*, $N(t_L)$ ($N(t_R)$) is the number of cases passed down to the left (right) child, 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 $\hat{y}_j=a_0+\sum_{s=1}^{m}a_sx_s$ is computed by combining the straight-line regression associated with the leaf t_L (t_R) with all univariate regression lines associated with regression nodes along the path from the root to t_L (t_R).

If X_i is discrete, SMOTI partitions attribute values into two sets, so that binary trees are always built. Partitioning is based on the same criterion applied in CART [1, pp. 247], which reduces the search for the best subset of categories from 2^{k-1} to k-1, where k is the number of distinct values for X_i.

The evaluation of the effectiveness of a regression step $Y=a+bX_i$ at node *t* cannot be naïvely based on the resubstitution error $R(t)$:

$$R(t)=\sqrt{\frac{1}{N(t)}\sum_{j=1}^{N(t)}(y_j-\hat{y}_j)^2}$$

where the estimator $\hat{y}_j$ is computed by combining the straight-line regression associated with *t* with all univariate regression lines associated with regression nodes along the path from the root to *t*. This would result in values of $\pi(X_i,Y)$ less than or equal to values of $\sigma(X_i,Y)$ for some splitting test involving X_i. Indeed, the splitting test "looks-ahead" to the best multiple linear regressions after the split on X_i is performed,

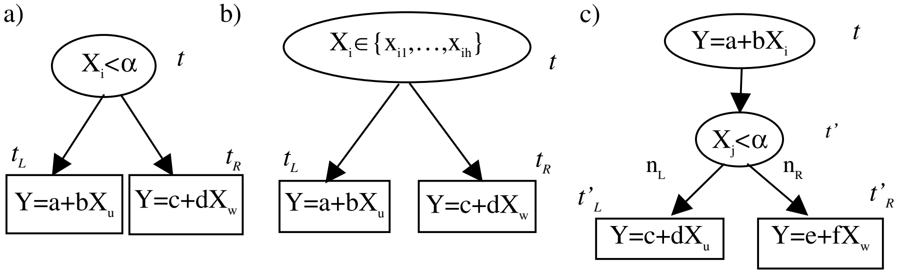

Fig. 1. a) A continuous split node. b) A discrete split node. c) An example of regression node.

while the regression step does not. A fairer comparison would be growing the tree at a further level in order to base the computation of $\pi(X_i,Y)$ on the best multiple linear regressions after the regression step on X_i is performed (see Figure 1c).

Let *t'* be the child of the regression node *t*, and suppose that it performs a splitting test. The best splitting test in *t'* can be chosen on the basis of $\sigma(X_j,Y)$ for all possible variables X_j, as indicated above. Then $\pi(X_i,Y)$ can be defined as follows:

$$\pi(X_i,Y) = min \{ R(t),\ \sigma(X_j,Y) \textit{ for all possible variables } X_j \}.$$

Having defined both $\pi(X_i,Y)$ and $\sigma(X_i,Y)$, the criterion for selecting the best node is fully characterized as well. A weight *w* (1-*w*) is associated with split (regression) nodes, so as to express the user preference for model trees with splitting tests (regression steps). Therefore, SMOTI actually compares the weighted values $w\sigma(X_i,Y)$ and $(1\text{-}w)\pi(X_i,Y)$ while selecting a node. At each step of the model tree induction process, SMOTI chooses the apparently most promising node according to a greedy strategy. A continuous variable selected for a regression step is eliminated from further consideration, so that it can appear only once in a regression node along a path from the root to a leaf.

In SMOTI three different stopping criteria are implemented. The first uses the partial F-test to evaluate the contribution to the model of a new independent variable [3]. The second requires the number of cases in each node to be greater than a minimum value. The third stops the induction process 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 computational complexity of the model tree induction algorithm is highly dependent on the choice of the best splitting test or regression step for a given node. For regression steps, the worst case complexity is $O(NmlogN)$, where N is the number of examples in the training set and m is the number of independent variables. For splitting tests, the worst case complexity is $O(N+NlogN)$, where the component $NlogN$ is due to the quicksort algorithm. Therefore, the worst case complexity for the selection of any node is $O(Nm^2 logN)$, since there are m independent variables.

It is noteworthy that SMOTI is more efficient than RETIS at building model trees and defines the best partitioning of the feature space coherently with respect to the model tree being built. Moreover, the use of both regression and split nodes permits the system to consider both global and local effects of variables in the various regression models. This is evident in the experimental results reported below.

4 Experimental Results and Discussion

SMOTI has been implemented as a module of the knowledge discovery system KDB2000 (www.di.uniba.it/~malerba/software/kdb2000/) and has been empirically evaluated on six datasets taken from UCI Machine Learning Repository (www.ics.uci.edu/~mlearn/MLRepository.html) and the site of the system HTL (www.ncc.up.pt/~ltorgo/Regression/DataSets.html). They are: a) *Abalone*, with 2889 cases and 8 attributes (7 continuous and 1 discrete); b) *Auto* with 398 cases and 8 attributes (5 continuous and 3 discrete); c) *Housing* with 506 cases and 14 continuous attributes;

d) *Machine CPU* with 209 cases and 6 discrete attributes; e) *Pyrimidines* with 74 cases and 27 continuous attributes; f) *Price* with 159 cases and 16 attributes (15 continuous and 1 discrete). Each dataset is analyzed by means of a 10-fold cross-validation, that is, the dataset is first divided into ten blocks of near-equal size and with near-equal distribution of class values, and then, for every block, SMOTI is trained on the remaining blocks and tested on the hold-out block. The system performance is evaluated on the basis of both the average resubstitution error and the average number of leaves. For pairwise comparison of methods, the non-parametric Wilcoxon signed rank test is applied [8], where the summations on both positive and negative ranks, namely W+ and W-, are used to determine the winner. In all experiments reported in this empirical study, the significance level α is set to 0.05.

4.1 Effect of Node Weighting

The first experiment aims at investigating the effect of node weighting on the predictive accuracy and complexity of the tree. A weight greater than 0.5 prefers splitting tests, while a weight lower than 0.5 favors the selection of regression nodes. It is noteworthy that, for higher weight values, regression nodes are often selected near the leaves of the tree, so that they can give only a local contribution to the approximation of the underlying function with a model tree. On the contrary, for lower values of the weight regression node they tend to be selected at the root, so that they give a global contribution to the approximation of the underlying function. In other words, the weight represents the trade-off between global regression models that span the whole feature space and are built using all training cases and local regression models, which fit fewer data falling in smaller portions of the feature space.

The weighting factor also affects the predictive accuracy of the induced model, as reported in Table 1. In each of the ten trials per dataset, predictive accuracy is estimated by the mean square error, computed on the corresponding validation set. Experimental results show that by increasing the weight, that is favoring the selection of split nodes, it is possible to obtain more accurate model trees. Moreover, we also observed that for weight values higher than 0.6 the situation does not change with respect to the case w=0.6, while for weight values lower than 0.5 the accuracy is lower than that observed with w=0.5. The conclusion is that, in almost all the data sets considered, local effects of regression variables are preferred.

Table 1. Results of the Wilcoxon signed rank test on the accuracy of the induced model. The best value is in boldface, while the statistically significant values ($p \leq \alpha/2$) are in italics

Data set	**0.5 vs 0.52**			**0.5 vs 0.56**			**0.5 vs 0.6**		
	p	*W+*	*W-*	*P*	*W+*	*W-*	*p*	*W+*	*W-*
Abalone	0.083	45	**10**	*0.004*	54	***1***	*0.004*	54	***1***
Auto	0.556	34	**21**	0.492	35	**20**	1.000	28	**27**
Housing	0.492	**20**	35	0.275	39	**16**	0.432	36	**19**
Machine	0.064	46	**9**	0.064	46	**9**	0.064	46	**9**
Price	0.083	45	**10**	0.232	**40**	15	0.432	36	**19**
Pyrimidines	*0.002*	55	***0***	0.106	11	**44**	0.064	46	**9**

SMOTI has also been compared to two other TDMTI systems, namely a trial version of Cubist and M5'. Experimental results, which are reported in [6] and are unfavorable to SMOTI, seem to confirm the presence of a common factor to many of the data sets used in the experiments on regression and model trees: no general behavior was noted for the underlying function to be approximated, and it can be better represented as a composition of many definite local behaviors.

4.2 Experiments on Artificial Data Sets

SMOTI was also tested on artificial data sets randomly generated for seven different model trees. These model trees were automatically built for learning problems with nine independent variables (five continuous and four discrete), where continuous variables take values in the unit interval [0,1], while discrete variables take values in the set {A,B,C,D,E,F,G}. The model tree building procedure is recursively defined on the maximum depth of the tree to be generated. The choice of adding a regression or a split node is random and depends on a parameter $\theta \in [0,100]$: the probability of selecting a split node is θ%; conversely, the probability of selecting a regression node is $(100-\theta)$%. Therefore, the returned model trees have a variable number of regression/split nodes and leaves, while the depth of the tree is kept under control. In the experiments reported in this paper θ is fixed to 0.5, while the depth is set to 5.

Ten data points are randomly generated for each leaf, so that the size of the data set associated with a model tree depends on the number of leaves in the tree itself. Data points are generated according to the different multiple linear models associated with the leaves. The error added to each model is distributed normally, with zero mean and variance σ^2, which is kept constant for all leaves. The value of σ^2 set for the experimentation is 0.001, which means that for almost 90% of generated data points the effect of the error is ±0.095, according to Chebyshev's inequality. It is noteworthy that the effect of the error is not marginal, given that both independent variables and their coefficients range in the unit interval.

Each dataset was analyzed by means of a 10-fold cross-validation. In order to study the effect of the weight, two different values were considered: *w*=0.5 and *w*=0.55. Experimental results are reported in Table 2. The number of leaves of the original model trees (T. #leaves) is compared to the corresponding property of the induced tree (denoted by the initial I).The last three columns list the average mean square error reported by SMOTI and M5´. Results show that SMOTI over-partitions the feature space, since the number of leaves in the induced trees is always greater than the number of leaves in the theoretical model tree. This is true even in the case of *w*=0.5. Interestingly, in many cases SMOTI outperforms M5´ with respect to average MSE.

Table 2. Results for the model tree built with parameters θ=0.5, depth=5, and σ^2=0.001.

T. depth	*T. # leaves*	*I. # leaves w=0.5*	*I. # leaves w=0.55*	*Av. MSE SMOTI w=0.5*	*Av. MSE SMOTI w=0.55*	*Av. MSE. M5'*
5	5	7	9	*0.24*	0.61	0.35
5	7	10	10	*0.2*	*0.15*	0.36
5	8	11	12	*0.19*	*0.17*	0.3
5	6	10	10	0.53	0.32	*0.27*
5	8	12	10	0.56	0.68	*0.24*
5	1	1	1	*0.16*	*0.16*	0.29
5	6	17	18	*0.15*	*0.16*	0.25

Results on a more extensive experimentation are reported in Table 3. They are obtained by keeping θ=0.5, σ^2=0.001, and by varying both the number of training cases per leaf (10, 20, 30 items) and the depth of the tree (5,6,7). Three main conclusions can be drawn from Table 4: first, SMOTI performs better than M5′ when split nodes are slightly preferred to regression nodes, that is, local decisions are favored; second, by increasing the number of training cases per leaf, no difference is observed in the trading-off between local and global effects[2]; third, the depth of the tree has no clear effect on the predictive accuracy of the induced model tree.

Table 3. Results of the Wilcoxon signed rank test on the accuracy of the induced model tree built with parameters θ=0.5 and σ^2=0.001. The best value is in boldface, while the statistically significant values ($p \leq \alpha/2$) are in italics.

Data set	**Depth**	**M5' vs. SMOTI (w=0.5)**			**M5' vs. SMOTI (w=0.55)**		
		P	***W+***	***W-***	***p***	***W+***	***W-***
10 items per leaf	5	0.937	15	**13**	1	14	14
	6	0.46	**9**	19	0.937	**13**	15
	7	*0.078*	***3***	25	1	14	14
20 items per leaf	5	*0.047*	***2***	26	0.6875	17	**11**
	6	*0.047*	***2***	26	*0.07812*	25	***3***
	7	0.93	**13**	15	0.2969	21	**7**
30 items per leaf	5	1	14	14	0.375	20	**8**
	6	0.218	**6**	22	0.5781	18	**10**
	7	0.687	**11**	17	*0.0312*	27	***1***

5 Conclusions

In the paper, a novel method, called SMOTI, has been presented. The main advantage of SMOTI is that it efficiently generates model trees with multiple regression models in the leaves. Model trees generated by SMOTI include two types of nodes: regression nodes and split node. A weight associated to the type of node permits the user to express a preference for either local regression or global regression.

Experimental results on UCI data sets proved that in most of them, local effects of regression variables are preferred. An empirical comparison with M5´ on artificial data sets proved that SMOTI could induce more accurate model trees when both global and local behaviors are mixed up in the underlying model. In the future, we plan to investigate the effect of pruning model trees. To date, no study on the simplification techniques for model trees has been presented in the literature. There are several possible approaches, some based on the direct control of tree size, and others based on the extension of the set of tests considered. Both a theoretical and an empirical evaluation of these approaches in terms of accuracy and interpretability would be helpful in practical applications.

[2] In a personal communication, Tom Mitchell hypothesized that the importance of taking into account "global" effects might vanish with larger training sets. This hypothesis is not evident in our results.

Acknowledgments. This work is part of the MURST COFIN-1999 project on "Statistical Models for Classification and Segmentation of Complex Data Structures: Methodologies, Software and Applications." The authors thank Valentina Tamma and Domenico Pallotta for their collaboration and Tom Mitchell for his valuable comments on a preliminary version of this paper.

References

1. Breiman L., Friedman J., Olshen R., & Stone J.: *Classification and regression tree*, Wadsworth & Brooks, 1984.
2. Ciampi A.: Generalized regression trees, *Computational Statistics and Data Analysis*, 12, pp. 57-78, 1991.
3. Draper N.R., & Smith H.: *Applied regression analysis*, John Wiley & Sons, 1982.
4. Karalic A.: Linear regression in regression tree leaves, in *Proceedings of ISSEK '92 (International School for Synthesis of Expert Knowledge)*, Bled, Slovenia, 1992.
5. Lubinsky D.: Tree Structured Interpretable Regression, in *Learning from Data*, Fisher D. & Lenz H.J. (Eds.), Lecture Notes in Statistics, 112, Springer, pp. 387-398, 1994.
6. Malerba D., Appice A., Bellino A., Ceci M., & Pallotta D.: Stepwise Induction of Model Trees. In F. Esposito (Ed.), *AI*IA 2001: Advances in Artificial Intelligence*, Lecture Notes in Artificial Intelligence,2175, Springer, Berlin, Germany, pp. 20-32, 2001.
7. Morgan J.N., & Sonquist J.A.: Problems in the analysis of survey data, and a proposal, in *American Statistical Association Journal*, pp. 415-434, 1963.
8. Orkin, M., Drogin, R.: *Vital Statistics*, McGraw Hill, New York (1990).
9. Quinlan J. R.: Learning with continuous classes, in Proceedings AI'92, Adams & Sterling (Eds.), World Scientific, pp. 343-348, 1992.
10. Siciliano R., & Mola F.: Modelling for recursive partitioning in variable selection, in *COMPSTAT '94*, Dutter R., & Grossman W. (Eds.), Physica-Verlag, pp. 172-177, 1994.
11. Torgo L.: Kernel Regression Trees, in *Poster Papers of the 9th European Conference on Machine Learning (ECML 97)*, M. van Someren, & G. Widmer (Eds.), Prague, Czech Republic, pp. 118-127, 1997.
12. Torgo L.: Functional Models for Regression Tree Leaves, in *Proceedings of the Fourteenth International Conference (ICML '97)*, D. Fisher (Ed.), Nashville, Tennessee, pp. 385-393, 1997.
13. 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, pp. 128-137, 1997.