Top-Down Induction of Model Trees with Regression and Splitting Nodes

Donato Malerba, *Member*, *IEEE*, Floriana Esposito, *Member*, *IEEE*,
Michelangelo Ceci, and Annalisa Appice

Abstract—Model trees are an extension of regression trees that associate leaves with multiple regression models. In this paper, a method for the data-driven construction 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 splitting nodes, which partition the feature space. The multiple linear model associated with each leaf is then built stepwise 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. Experimental results on artificially generated data sets show that SMOTI outperforms two model tree induction systems, M5' and RETIS, in accuracy. Results on benchmark data sets used for studies on both regression and model trees show that SMOTI performs better than RETIS in accuracy, while it is not possible to draw statistically significant conclusions on the comparison with M5'. Model trees induced by SMOTI are generally simple and easily interpretable and their analysis often reveals interesting patterns.

Index Terms—Inductive learning, linear regression, model trees, global and local effects, regression and splitting nodes, SMOTI.

1 Introduction

Many problems encountered in common practice involve the prediction of a continuous numeric attribute associated with a case. More formally, given a set of observed data $(\mathbf{x}, \mathrm{y}) \in \mathbf{X} \times \mathrm{Y}$, where $\mathbf{X}$ denotes the feature space spanned by m independent (or predictor) variables x_i (both numerical and categorical), the goal is to predict the dependent (or response) variable Y which is continuous. This problem has been approached in many ways, such as standard regression, neural nets, and regression trees [1]. A *regression tree* approximates a function $\mathrm{y} = \mathrm{g}(\mathbf{x})$ by means of a piecewise *constant* function. *Model trees* generalize the concept of regression trees in the sense that they approximate $\mathrm{g}(\mathbf{x})$ by a piecewise *linear* function, that is, they associate leaves with multiple linear models. The problem of inducing model trees from a training set has received attention both in statistics and in machine learning. Some of the model tree induction systems developed are: M5 [18], RETIS [9], M5' [23], TSIR [13], HTL [20], which has been subsequently included in RT [21], SUPPORT [2], which has been extended in GUIDE [12] and SECRET [3].

All these systems perform a *top-down* induction of model trees (TDIMT) by recursively partitioning the training set. However, some of them (e.g., M5, M5′, and HTL) first build the tree structure and *then* associate leaves with linear models. In this way, the heuristic evaluation function used to select the best partition is computationally efficient, but it may compromise the discovery of the "correct" trees because of its incoherence with the linear model associated with the leaves. A different approach is followed in SUPPORT and SECRET, which reduce the computational complexity by transforming a regression problem into a classification problem and then by choosing the best partition on the basis of computationally efficient evaluation functions developed for classification tasks.

Another common characteristic of almost all these TDIMT systems is that the multiple regression 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. To explain this concept, let us consider the case of a region R of a feature space described by four continuous independent variables X_1, X_2, X_3, and X_4. The region R can be partitioned into two regions, R_1 and R_2, such that cases with $X_4 \leq \alpha\, (X_4 > \alpha)$, for a constant threshold α, fall in R_1 (R_2). Two regression models can be built independently for each region R_i, say:

$$R_1 : \hat{Y} = b'_0 + b'_1 X_1 + b'_2 X_2 \tag{1}$$

$$R_2 : \hat{Y} = b''_0 + b''_1 X_1 + b''_3 X_3. \tag{2}$$

The presence of X_1 in both models simply indicates that this variable is relevant both when $X_4 \leq \alpha$ and when $X_4 > \alpha$, although its influence on the dependent variable Y can be very different for the two regions. In this case, we say that the effect of X_1 on Y is *local* since it can be properly predicted by considering the test $X_4 \leq \alpha$. A *global* effect occurs when the contribution of X_1 to Y can be reliably predicted on the whole region R. In this case, an initial regression model can be approximated by regressing on X_1 for the whole region R:

- *The authors are with the Dipartimento di Informatica, Università degli Studi, via Orabona, 4, I-70125 Bari, Italy.
E-mail: {malerba, esposito, ceci, appice}@di.uniba.it.*

*Manuscript received 4 Feb. 2003; revised 26 Aug. 2003; accepted 30 Nov. 2003.
Recommended for acceptance by T. Tan.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 118227.*

$$R : \hat{Y} = b_0 + b_1 X_1 \tag{3}$$

and then by adding the effect of the variables X_2 and X_3 locally to the subregions R_1 and R_2, respectively. This is a case of stepwise construction of regression models [4]. As explained later, the correct procedure to follow in order to introduce the effect of another variable in the partially constructed regression model is to eliminate the effect of X_1. In practice, this means that we have to compute the following regression models for the whole region R:

$$R : \hat{X}_2 = b_{20} + b_{21} X_1 \tag{4}$$

$$R : \hat{X}_3 = b_{30} + b_{31} X_1 \tag{5}$$

and then to compute the residuals $X_2' = X_2 - \hat{X}_2, X_3' = X_3 - \hat{X}_3$ and $Y' = Y - \hat{Y} = Y - (b_0 + b_1 X_1)$. By regressing the residuals Y' on X_2' and X_3' for the regions R_1 and R_2, respectively, the following two models are built:

$$\begin{aligned} R_1 : \hat{Y} &= b_0 + b_1 X_1 + b_2 X_2' \\ &= b_0 + b_1 X_1 + b_2 X_2 - b_2 b_{20} - b_2 b_{21} X_1 \end{aligned} \tag{6}$$

$$\begin{aligned} R_2 : \hat{Y} &= b_0 + b_1 X_1 + b_3 X_3' \\ &= b_0 + b_1 X_1 + b_3 X_3 - b_3 b_{30} - b_3 b_{31} X_1. \end{aligned} \tag{7}$$

They show the *global* effect of X_1 since their first two common terms do not depend on the test $X_4 \leq \alpha$, that is, they are computed on the whole region R. Moreover, the last term of each model corrects the contribution of X_1 due to the *local* introduction of either X_2 or X_3.

In model trees, global effects can be represented by variables that are introduced in the multiple models at higher levels of the tree. 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 with the leaves.

This paper, which extends and revises the work in [14], presents the current state of the art on TDIMT and starting from the strengths and weaknesses of some approaches, proposes a new method, called *S*tepwise *Mo*del *T*ree *I*nduction (SMOTI), that constructs model trees stepwise, by adding, at each step, either a regression node or a splitting node. Regression nodes perform straight-line regression, while splitting nodes partition the feature 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. 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. In addition to solving the problem of modeling phenomena where some variables have a global effect while others have only a local effect, the stepwise construction supported by SMOTI permits at no additional cost to define a heuristic evaluation function which is coherent with the linear models at the leaves.

The paper is organized as follows: The state of the art model tree induction is described in the next section, while, in Section 3 the SMOTI method is introduced and its computational complexity is analyzed. In Section 4, some experimental results are reported for both artificially generated data and data typically used in the evaluation of regression and model trees. For this second set of experimental results, the detected presence of some global effects is also discussed.

2 Background and Motivation

In the top-down construction of a model tree, one of the main problems to be solved is choosing the best partition of a region of the feature space. Several evaluation functions have been proposed. 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:

$$R(T) = \frac{1}{N} \sum_{t \in \tilde{T}} \sum_{x_i \in t} (y_i - \bar{y}(t))^2, \tag{8}$$

where N is the number of training examples $(\mathbf{x}_i, \mathrm{y}_i)$, $\tilde{T}$ 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 $s^2(t)$ the sample variance at a node t, $R(T)$ can be rewritten as:

$$R(T) = \sum_{t \in \tilde{T}} \frac{N(t)}{N} s^2(t) = \sum_{t \in \tilde{T}} p(t) s^2(t), \tag{9}$$

where $N(t)$ is the number of observations in the node t and $p(t)$ is the estimated 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 $R(T')$. More precisely, the minimization of $R(T')$ is equivalent to minimizing $p(t_L)s^2(t_L) + p(t_R)s^2(t_R)$, which is the contribution to $R(T')$ provided by the split.

This heuristic criterion, initially conceived for a regression tree, has also been used for model trees. In the HTL system, 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*.

To illustrate the problem, let us consider the data set plotted in Fig. 1a and generated according to the model tree in Fig. 1b. Neither M5 nor its commercial version, named Cubist, nor HTL are able to find the underlying model tree because of net separation of the *splitting* stage from the *predictive* one and, in particular, due to the fact that the partitioning of the feature space does not take into account the regression models that can be associated with the leaves. This seems to be inherited by regression tree learners, such as CART. In this case, however, the evaluation functions (e.g., $R(T)$) do take into account the models built in the leaves (the sample means). On the contrary, when we try to use the same heuristic criteria for model tree induction, we are rating the effectiveness of a partition with respect to different models from the ones chosen in the subsequent predictive stage.

This 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

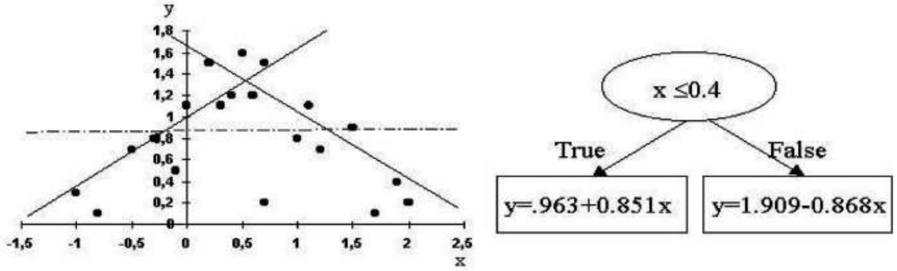


Fig. 1. (a) Scatter plot of 20 cases; the values of the only independent variable range between -1.0 and 2.0. A simple linear regression on the whole data set would give the dashed line. (b) The underlying model tree partitions the training cases into two subgroups: $X \leq 0.4$ and $X > 0.4$.

error with respect to the regression plane g_L (g_R) found for the left (right) child:

$$s^2(t_L) = \frac{1}{N(t_L)} \sum_{x_i \in t_L} (y_i - g_L(x_i))^2$$
$$\left(s^2(t_R) = \frac{1}{N(t_R)} \sum_{x_i \in t_R} (y_i - g_R(x_i))^2 \right). \tag{10}$$

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. However, the computational complexity of this evaluation function is cubic in the number of independent continuous[1] variables, m, and quadratic in the number of training observations, N. Indeed, the selection of the first split takes time $O(mNlogN)$ to sort all values of the m variables, plus time required to test $(N-1)m$ distinct cut points, at worst. Each test, in turn, requires the computation of two regression planes on the m independent variables, that is, twice the time of computing $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$, where $\mathbf{y}$ is the $N(t)$-dimensional vector of values taken by the response variable in node t, while $\mathbf{X}$ is an $N(t)(m+1)$ matrix of observations plus a column with only 1s [4]. Solving this least-square problem by normal equations takes time $O(N(t)(m+1)^2 + (m+1)^3)$ since, in general, $N(t) > m$, the time complexity can be approximated to $O(N(t)(m+1)^2)$. For at least one of the tests, $N(t)$ is proportional to N, thus the choice of the first split takes time $O(N(N-1)m(m+1)^2)$.

In SUPPORT and SECRET, this inefficiency is solved by transforming a regression problem into a classification problem. More precisely, Chaudhuri et al. [2] propose labeling the residuals of a linear model at a node as either positive or negative and choosing the best partition on the basis of computationally efficient statistical tests developed for classification tasks. The justification of this approach is that, if a fitted model is unsatisfactory, the lack of fit would be reflected in the distributional pattern of the residuals. Dobra and Gehrke [3] observe that this justification is not theoretically founded and propose identifying two normal multivariate distributions in the space $\mathbf{X} \times Y$ and then classifying observed data according to the probability of belonging to these two distributions. The best partition is selected by means of efficient techniques developed for decision trees. In the case of binary splits of continuous attributes, Torgo [22] proposes a solution based on a recursive least squares algorithm whose average complexity is quadratic in the number of different data points between two subsequent cut-points of the continuous variable.

In addition to high computational complexity, another problem may occur in RETIS since the regression planes g_L and g_R involve all continuous variables. When some of the independent variables are related to each other, that is, they are (approximately) collinear, several problems may occur [4], such as indeterminacy of regression coefficients, unreliability of the estimates of the regression coefficients, and impossibility of evaluating the relative importance of the independent variables. Interestingly, problems due to collinearity do not show in the model's fit. The resulting model may have very small residuals, but the regression coefficients are actually poorly estimated. A treatment suggested for data that exhibit collinearity is that of deleting some of the variables from a fitted model. Therefore, variable subset selection is a desirable part of regression analysis.

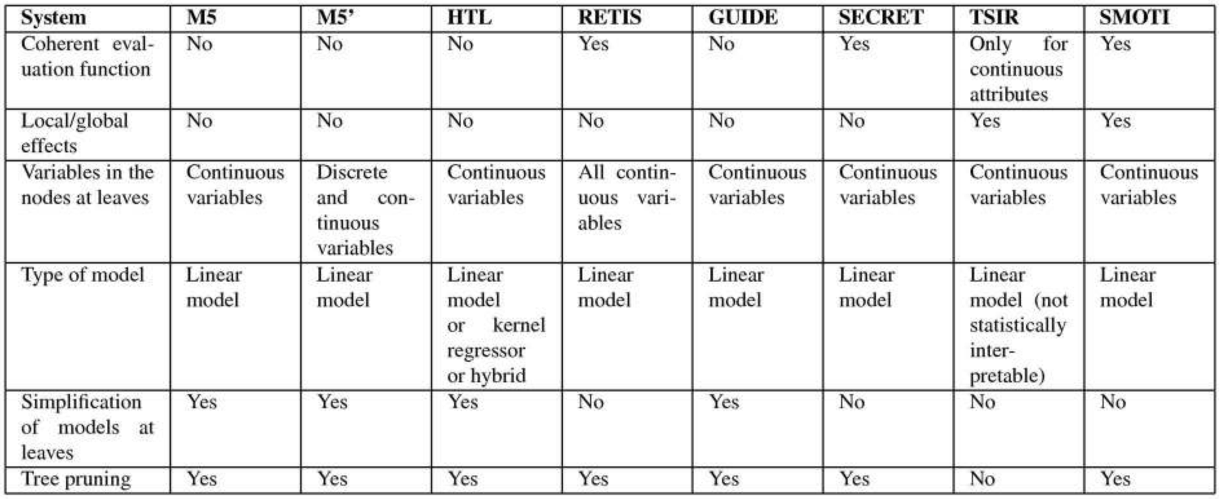
Finally, RETIS, as well as many other TDIMT systems, is characterized by models at leaves that can take into account only local decisions, as explained in Section 1. A solution to this problem is the stepwise construction of multiple linear models by intermixing regression steps with partitioning steps, as in TSIR. TSIR has two types of node: splitting nodes and regression nodes. A splitting node performs a Boolean test on a variable and has two children. A regression node computes a single variable regression, $\hat{Y} = a + bX$, and passes 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." 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* [4]. For instance, let us consider the problem of building a multiple regression model with two independent variables through a sequence of straight-line regressions: $\hat{Y} = a + bX_1 + cX_2$. We start regressing Y on X_1 so that the model:

$$\hat{Y} = a_1 + b_1 X_1$$

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

[1] The complexity for discrete variables cannot be evaluated since no specification is reported in the literature on the procedure that RETIS follows to select best subsets of attribute values.

TABLE 1
Systems Comparison



System	M5	M5'	HTL	RETIS	GUIDE	SECRET	TSIR	SMOTI
Coherent evaluation function	No	No	No	Yes	No	Yes	Only for continuous attributes	Yes
Local/global effects	No	No	No	No	No	No	Yes	Yes
Variables in the nodes at leaves	Continuous variables	Discrete and continuous variables	Continuous variables	All continuous variables	Continuous variables	Continuous variables	Continuous variables	Continuous variables
Type of model	Linear model	Linear model	Linear model or kernel regressor or hybrid	Linear model	Linear model	Linear model	Linear model (not statistically interpretable)	Linear model
Simplification of models at leaves	Yes	Yes	Yes	No	Yes	No	No	No
Tree pruning	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes

both X_1 and X_2, we can build a linear model for X_2 if X_1 is given: $\hat{X}_2 = a_2 + b_2 X_1$, then compute the residuals on X_2 and Y:

$$X_2' = X_2 - (a_2 + b_2 X_1); Y' = Y - (a_1 + b_1 X_1)$$

and, finally, regress Y' on X_2' alone: $\hat{Y}' = a_3 + b_3 X_2'$.

By substituting the equations of X_2' and Y' in the last equation, we have:

$$Y - (\widehat{a_1 + b_1 X_1}) = a_3 + b_3(X_2 - (a_2 + b_2 X_1)).$$

Since $Y - (\widehat{a_1 + b_1 X_1}) = \hat{Y} - (a_1 + b_1 X_1)$, we have:

$$\hat{Y} = (a_3 + a_1 - a_2 b_3) + (b_1 - b_2 b_3) X_1 + b_3 X_2.$$

It can be proven that this last model coincides with the first model built, that is, $a = a_3 + a_1 - a_2 b_3$, $b = b_1 - b_2 b_3$, and $c = b_3$. Therefore, when the first regression line of Y on X_1 is built, we pass down both the residuals of Y and *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 both the response variable (Y) and those variables to be selected for the next regression step (X_2). TSIR operates in a different way since it passes down the residuals of Y alone. Therefore, 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. Moreover, collinearity problems are not properly solved, although only a subset of variables may be involved in the models at leaves.

A summary of some characteristics discussed above is reported in Table 1. It is noteworthy that all systems can build multiple linear regression models at leaves. In addition, HTL can build kernel regressors, which simply implement but do not capture the structure of the domain as linear models do. Some systems involve both continuous and discrete variables in the linear models at leaves. The latter are treated as dichotomous variables in standard linear regression [4]; however, their real contribution is unclear in the case of model trees. In some systems, linear models at leaves can be retrospectively simplified by deleting some variables.

In the next section, the new TDIMT system SMOTI is presented. It has four distinguishing features:

1. A selection measure is chosen which is coherent with respect to the (partial) linear model associated with the leaves.
2. Multiple regression models are constructed stepwise by intermixing both regression and splitting nodes. Problems observed in TSIR are solved by removing the effect of the variable selected in a regression node before passing down training cases to deeper levels and by adopting a look-ahead strategy when regression nodes and splitting nodes are compared for selection.
3. The multiple linear model associated with each leaf involves all the numerical variables in the regression nodes and the numerical variable in the straight-line regression performed at the leaf. In this way, both global and local effects of variables are considered.
4. The simplification strategies apply only to the tree structure, thus the deletion of some variables in the models at leaves is the result of pruning a regression node.

The description of the simplification methods and the related experimental results are not presented because of space constraints. In this paper, the presentation is focused only on the first three distinguishing features.

3 Stepwise Construction of Model Trees

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 nodes in the tree: regression nodes and splitting nodes. 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 and no change is made on the variables. For a regression node t, all the observations are passed down to its only child, but both the values of the dependent variable and the values of the (continuous) independent variables

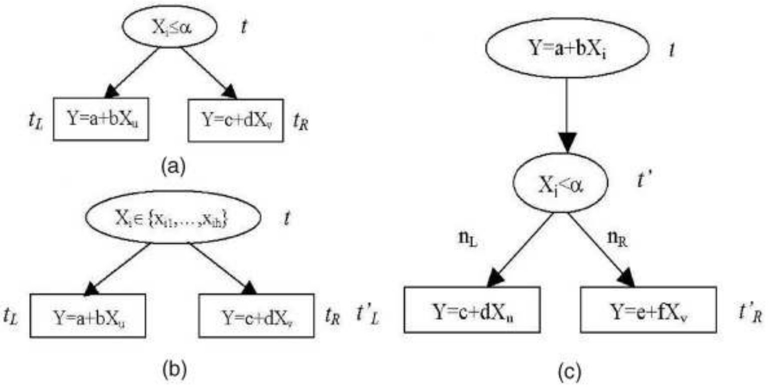


Fig. 2. (a) A continuous split node t with two straight-line regression models in the leaves. (b) A discrete split node t with two straight-line regression models in the leaves. (c) Evaluation of a regression step at node t, based on the best splitting test below.

not included in the model are transformed to remove the linear effect of those variables already included. This is done coherently with the statistical theory for the incremental construction of a multiple linear regression model, as explained in Section 2. Thus, descendants of a regression node will operate on a modified data set.

The validity of either a regression step or a splitting test on a variable X_i is based on two distinct evaluation measures, $\rho(X_i, Y)$ 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 $\rho(X_i, Y)$ and $\sigma(X_i, Y)$ are mean square errors,[2] therefore, they can actually be compared to choose between three different possibilities:

1. growing the model tree by adding a regression node t;
2. growing the model tree by adding a splitting node t;
3. stopping the tree's growth at node t.

The evaluation measure $\sigma(X_i, Y)$ should be coherently defined on the basis of the multiple linear models at the leaves. In SMOTI, it is sufficient to consider the best straight-line regression associated to each leaf t_R (t_L) since regression nodes along the path from the root to t_R (t_L) already partially define a multiple regression model (see Figs. 2a and 2b).

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

$$\sigma(X_i, Y) = \frac{N(t_L)}{N(t)} R(t_L) + \frac{N(t_R)}{N(t)} R(t_R), \tag{11}$$

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:

$$\begin{gathered} R(t_L) = \sqrt{\frac{1}{N(t_L)} \sum_{x_j \in t_L} (y_j - \hat{y}_j)^2} \\ \left(R(t_R) = \sqrt{\frac{1}{N(t_R)} \sum_{x_j \in t_R} (y_j - \hat{y}_j)^2} \right). \end{gathered} \tag{12}$$

The estimate:

$$\hat{y}_j = a_0 + \sum_s a_s x_s \tag{13}$$

is computed by combining all univariate regression lines associated with regression nodes along the path from the root to t_L (t_R). Possible values of α are found by sorting the distinct values of X_i in the training set associated to t, then identifying a threshold between each pair of adjacent values. Therefore, if the cases in t have k distinct values for X_i, $k-1$ thresholds are considered. Obviously, the lower $\sigma(X_i, Y)$, the better the split $X_i \leq \alpha$.

If X_i is discrete, SMOTI partitions attribute values into two sets so that binary trees are always built. Some TDIMT systems, such as HTL and M5′, use the same criterion applied in CART [1, p. 247]. More precisely, if k is the number of distinct values for X_i and $S_{X_i} = \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$ is the set of distinct values of X_i, S_{X_i} is sorted according to the sample mean of Y over all cases in t. A theorem by Breiman et al. [1] (Theorem 4.5, Proposition 8.16) proves that the best binary split is one of $k-1$ partitions $\{x_{i_1}, \ldots, x_{i_h}\}$ and $S_{X_i} - \{x_{i_1}, \ldots, x_{i_h}\}$, thus greatly reducing the search for the best subset of categories from 2^{k-1} to $k-1$ partitions. However, the theorem is based on the assumption that the models at the leaves are the sample means, which is not the case of SMOTI.[3] Therefore, SMOTI relies on a nonoptimal greedy strategy as suggested by [16]. It starts with an empty set $Left_{X_i} = \phi$ and a full set $Right_{X_i} = S_{X_i}$. It moves one element from $Right_{X_i}$ to $Left_{X_i}$ such that the move results in a better split. The evaluation measure $\sigma(X_i, Y)$ is computed as in the case of continuous variables and, therefore, a better split decreases $\sigma(X_i, Y)$. The process is iterated until there is no improvement in the splits. The computational complexity of this heuristic is O(k^2). For all possible splits, the measure $\sigma(X_i, Y)$ is computed as in the case of continuous variables.

The split selection criterion explained above can be improved to consider the special case of identical regression model associated with both children. When this occurs, the best straight-line regression associated with t is the same as that associated with both t_L and t_R, up to some statistically insignificant difference. In other terms, the split is useless and can be filtered out from the set of alternatives. To check this special case, SMOTI compares the two regression lines

[2] This is different from TSIR, which, in the case of node selection, minimizes the absolute deviation between a *constant* value (the median) and the observed values Y. On the contrary, SMOTI coherently minimizes the square error with respect to the partially constructed regression model at each node.

[3] Sample means are used only when all independent variables are discrete.

associated with the children according to a statistical test for coincident regression lines [24, pp. 162-167].

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

$$R(t) = \sqrt{\frac{1}{N(t)}\sum_{j=1}^{N(t)} (y_j - \hat{y}_j)^2}, \tag{14}$$

where the estimator $\hat{y}_j$ is computed by combining all univariate regression lines associated with regression nodes along the path from the root to t. This would result in values of $\rho(X_i, Y)$ 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, 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 $\rho(X_i, Y)$ on the best multiple linear regressions after the regression step on X_i is performed (see Fig. 2c).

Let t' be the child of the regression node t and let us 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, $\rho(X_i, Y)$ can be defined as follows:

$$\rho(X_i, Y) = min\{R(t), \sigma(X_j, Y) for\ all\ possible\ variables X_j\}. \tag{15}$$

The possibility of statistically identical regression models associated with the children of t' may also occur in this case. When this happens, the splitting node is replaced by another regression node t' where the straight-line regression model is the same as that in the children of the splitting node. Therefore, in this special case, $\rho(X_i, Y)$ can be defined as follows:

$$\rho(X_i, Y) = min\{R(t), R(t')\}. \tag{16}$$

Having defined both $\rho(X_i, Y)$ and $\sigma(X_i, Y)$, the criterion for selecting the best node is fully characterized as well. 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 no longer considered for regression purposes so that it can appear only once in a regression node along a path from the root to a leaf.

In SMOTI, five different stopping criteria are implemented. The first uses the partial F-test to evaluate the contribution of a new independent variable to the model [4]. 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 fourth creates a leaf if the error in the current node is below a fraction of the error in the root node, as in [21, p. 60]. Finally, the fifth stops the induction process when the coefficient of determination is greater than a minimum value [24, pp. 18-19]. This coefficient is a scale-free one-number summary of the strength of the relationship between independent variables in the actual multiple model and the response variable.

The computational complexity of adding a splitting node t to the tree depends on the complexity of a splitting test selection in t multiplied by the complexity of the best regression step selection in the children nodes t_R and t_L. On the contrary, the computational complexity of adding a regression node t depends on the complexity of a regression step selection in t multiplied by the complexity of the best splitting test in its child t'.

A splitting test can be either continuous or discrete. In the former case, a threshold α has to be selected for a continuous variable. Let N be the number of examples in the training set, then the number of distinct thresholds can be $N - 1$ at worst. They can be determined after sorting the set of distinct values. If m is the number of independent variables, the determination of all possible thresholds has a complexity $\mathrm{O}(mNlogN)$ when an optimal algorithm is used to sort the values. For each of the $m(N - 1)$ thresholds, SMOTI finds the best straight-line regression at both children, which has a complexity of $m(N - 1)$ in the worst case. Therefore, the splitting test has a complexity $\mathrm{O}(mNlogN + m^2(N - 1)^2)$, that is, $\mathrm{O}(m^2N^2)$. Similarly, for a discrete splitting test, the worst-case complexity is $\mathrm{O}(mk^2)$, where k is the maximum number of distinct values of a discrete variable. The selection of the best discrete splitting test has a complexity $\mathrm{O}(m^2k^2N)$. Therefore, finding the best splitting node (either continuous or discrete) has a complexity $O(m^2N^2 + m^2k^2N)$ and, under the reasonable assumption that $k^2 \leq N$, that is, the number of distinct values of the a discrete variable is less then the square root of the number of cases, the worst case complexity is $O(m^2N^2)$.

The selection of the best regression step requires the computation, for each of the m variables, of m straight-line regressions (one for the regression node plus $m - 1$ to remove the effect of the regressed variable) and the updating of the data set. This takes time $O(m(mN + mN))$ since the complexity of the computation of a straight-line regression is linear in N. Moreover, for each straight-line regression, a splitting test is required, which has a worst-case complexity of $O(m^2N^2)$. Therefore, the selection of the best regression step has a complexity $O(m^2N + m^3N^2)$, that is, $O(m^3N^2)$.

The above results lead to an $O(m^3N^2)$ worst case complexity for the selection of any node (splitting or regression). This means that, relative to node selection, SMOTI has the same complexity as RETIS but is less efficient than TSIR which adopts a v-fold cross-validation strategy without look-ahead for regression and splitting nodes. In TSIR, the complexity is $O(mvN)$ for regression nodes and $\mathrm{O}(mvN^2)$ for splitting nodes. However, the model that TSIR considers at the children of a discrete splitting node during its evaluation is the sample mean and not a linear regression, which means that it suffers from the problems of adopting a heuristic evaluation function which is not coherent with the models associated to the leaves.

In conclusion, SMOTI presents several advantages. First, it defines the best partitioning of the feature space coherently with respect to the model tree being built. Second, it provides a solution to the problems of collinearity at the same computational cost of RETIS. Third, the use of both regression and splitting nodes permits the system to discover both global and local effects of variables in the various regression models. This is evident in the experimental results reported below.

4 An Empirical Evaluation of SMOTI

SMOTI has been implemented as a module of the knowledge discovery system KDB2000 (http://www.di.uniba.it/~malerba/software/kdb2000/) and has been empirically evaluated both on artificially generated data and on data sets typically used in the evaluation of regression and model trees. Each data set is analyzed by means of a 10-fold cross-validation. The system performance is evaluated on the basis of the average mean square error (MSE):

$$AvgMSE = \frac{1}{k}\sum_{v\in V}\sqrt{\frac{1}{N(\bar{v})}\sum_{j\in v}(y_j - \hat{y}_j(\bar{v}))^2}, \qquad (17)$$

where $V = \{v_1, \ldots, v_k\}$ is a cross-validation partition, each v_i is a set of indices of training cases, k is the number of folds (i.e., 10), $N(\bar{v})$ is the number of cases in $V - v$, and $\hat{y}_j(\bar{v})$ is the value predicted for the $j\mathrm{th}$ training case by the model tree built from $V - v$.

For pairwise comparison of methods, the nonparametric Wilcoxon two-sample paired signed rank test is used [17] since the number of folds (or "independent" trials) is relatively low and does not justify the application of parametric tests, such as the t-test. To perform the test, we assume that the experimental results of the two methods compared are independent pairs of sample data $\{(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)\}$. We then rank the absolute value of the differences $u_i - v_i$. The Wilcoxon test statistics W^+ and W^- are the sum of the ranks from the positive and negative differences, respectively. We test the null hypothesis H_0: "no difference in distributions," against the two-sided alternative H_a: "there is a difference in distributions." More formally, the hypotheses are: H_0: "$\mu_u = \mu_v$" against H_a: "$\mu_u \neq \mu_v$." Intuitively, when $W^+ \gg W^-$ and vice versa, H_0 is rejected. Whether W^+ should be considered "much greater than" W^- depends on the significance level α. The basic assumption of the statistical test is that the two populations have the same continuous distribution (and no ties occur). Since, in our experiments, u_i and v_i are MSE, $W^+ \gg W^-$ implies that the second method (V) is better than the first (U). In all experiments reported in this empirical study, the significance level α used in the test is set at 0.05.

SMOTI has been compared to both M5′, which is considered the state-of-the-art model tree induction system, and RETIS, which has an evaluation function coherent with the models at the leaves.[4] The empirical comparison with TSIR, which is the only other system with regression and splitting nodes, was not possible since the system is not publicly available.

4.1 Experiments on Artificial Data Sets

SMOTI was initially tested on artificial data sets randomly generated for model trees with both regression and splitting nodes. These model trees were automatically built for learning problems with nine independent variables (five continuous and four discrete) where 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 splitting node is random and depends on a parameter $\theta \in [0, 1]$: The probability of selecting a splitting node is θ; conversely, the probability of selecting a regression node is $(1 - \theta)$. In the experiments reported in this paper, θ is fixed at 0.5, while the depth varies from four to nine. Fifteen model trees are generated for each depth value, for a total of 90 trees.

Sixty 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 by considering the various constraints associated with both splitting nodes and regression nodes. In the case of a splitting node, the only constraint is that the distribution of cases between left and right children should take into account the number of leaves in each subtree. In the case of a regression node, the constraints are the (partial) multiple linear model associated with the node, as well as the linear models defined for the residuals of the variables passed down. The noise effect is introduced by adding a normally distributed error $\sim N(0, 1)$ to the linear models relating independent variables and $\sim N(0, .001)$ to the linear models at the leaves involving the dependent variable. In all experiments, the thresholds for stopping criteria 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 percent of the error in the root node, the coefficient of determination in each internal node must be below 0.99.

In Table 2, the results of the test on the accuracy of trees induced by SMOTI, M5′, and RETIS are reported. Three main conclusions can be drawn from these experimental results: First, SMOTI performs generally better than M5′ and RETIS on data generated from model trees where both local and global effects can be represented. Second, by increasing the depth of the tree, SMOTI tends to be more accurate than M5′ and RETIS. Third, when SMOTI performs worse than M5′ and RETIS, this is due to relatively few hold-out blocks in the cross validation so that the difference is never statistically significant in favor of M5′ or RETIS.

An example of different results provided by SMOTI and M5′ is reported in Fig. 3. The underlying model tree, according to which a data set of 180 cases is generated, is reported in Fig. 3a. It first partitions the feature space into two subregions:

$$R_1 : \{(\mathbf{X}, Y) | X_5 \in \{0\}\}; R_2 : \{(\mathbf{X}, Y) | X_5 \in \{1, 2, 3, 4, 5, 6, 7\}\}.$$

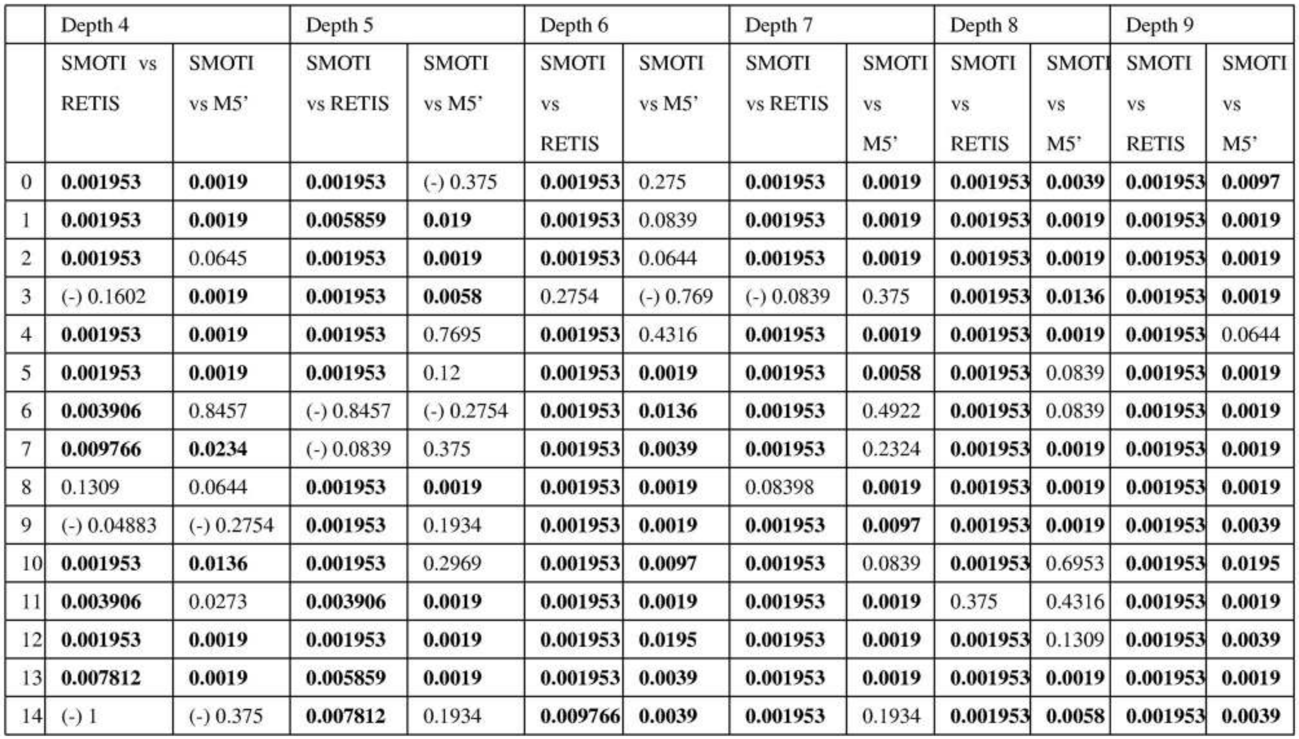
The subregion R_1 is in turn partitioned into two subregions:

$$R_{11} : \{(\mathbf{X}', Y') | X_6 \in \{0, 1, 2, 3, 4\}\};$$
$$R_{12} : \{(\mathbf{X}', Y') | X_6 \in \{5, 6, 7\}\}.$$

The variable X_1, which contributes to the regression models associated with both R_{11} and R_{12}, has a global effect on the response variable Y since its coefficient can be reliably estimated on the region R_1. On the contrary, the variables X_0 and X_2 have a local effect since their contributions to the regression models at the leaves can be estimated on the basis of the cases falling in the subregions R_{11}, R_{12}, and R_2 associated with the leaves. Actually, straight-line regressions at the leaves involve variables X_0', X_2', and X_0'', which are obtained by removing the effect of other variables already introduced in the model. It is noteworthy that the intercepts of straight-line regressions associated with nodes below a regression node are all equal to zero since we are using sets of

[4] When running M5′, the pruning factor (parameter $-F$) is set to 0 since the evaluation of the pruning effects in model tree induction is beyond the scope of this work. For the same reason, the pruning function is not invoked in RETIS. All remaining parameters are set to default values.

TABLE 2
SMOTI versus M5' and RETIS: Results of the Wilcoxon Signed Rank Test on the Accuracy of the Induced Model Trees



	Depth 4		Depth 5		Depth 6		Depth 7		Depth 8		Depth 9	
	SMOTI vs RETIS	SMOTI vs M5'	SMOTI vs RETIS	SMOTI vs M5'	SMOTI vs RETIS	SMOTI vs M5'	SMOTI vs RETIS	SMOTI vs M5'	SMOTI vs RETIS	SMOTI vs M5'	SMOTI vs RETIS	SMOTI vs M5'
0	**0.001953**	**0.0019**	**0.001953**	(-) 0.375	**0.001953**	0.275	**0.001953**	**0.0019**	**0.001953**	**0.0039**	**0.001953**	**0.0097**
1	**0.001953**	**0.0019**	**0.005859**	**0.019**	**0.001953**	0.0839	**0.001953**	**0.0019**	**0.001953**	**0.0019**	**0.001953**	**0.0019**
2	**0.001953**	0.0645	**0.001953**	**0.0019**	**0.001953**	0.0644	**0.001953**	**0.0019**	**0.001953**	**0.0019**	**0.001953**	**0.0019**
3	(-) 0.1602	**0.0019**	**0.001953**	**0.0058**	0.2754	(-) 0.769	(-) 0.0839	0.375	**0.001953**	**0.0136**	**0.001953**	**0.0019**
4	**0.001953**	**0.0019**	**0.001953**	0.7695	**0.001953**	0.4316	**0.001953**	**0.0019**	**0.001953**	**0.0019**	**0.001953**	0.0644
5	**0.001953**	**0.0019**	**0.001953**	0.12	**0.001953**	**0.0019**	**0.001953**	**0.0058**	**0.001953**	0.0839	**0.001953**	**0.0019**
6	**0.003906**	0.8457	(-) 0.8457	(-) 0.2754	**0.001953**	**0.0136**	**0.001953**	0.4922	**0.001953**	0.0839	**0.001953**	**0.0019**
7	**0.009766**	**0.0234**	(-) 0.0839	0.375	**0.001953**	**0.0039**	**0.001953**	0.2324	**0.001953**	**0.0019**	**0.001953**	**0.0019**
8	0.1309	0.0644	**0.001953**	**0.0019**	**0.001953**	**0.0019**	0.08398	**0.0019**	**0.001953**	**0.0019**	**0.001953**	**0.0019**
9	(-) 0.04883	(-) 0.2754	**0.001953**	0.1934	**0.001953**	**0.0019**	**0.001953**	**0.0097**	**0.001953**	**0.0019**	**0.001953**	**0.0039**
10	**0.001953**	**0.0136**	**0.001953**	0.2969	**0.001953**	**0.0097**	**0.001953**	0.0839	**0.001953**	0.6953	**0.001953**	**0.0195**
11	**0.003906**	0.0273	**0.003906**	**0.0019**	**0.001953**	**0.0019**	**0.001953**	**0.0019**	0.375	0.4316	**0.001953**	**0.0019**
12	**0.001953**	**0.0019**	**0.001953**	**0.0019**	**0.001953**	**0.0195**	**0.001953**	**0.0019**	**0.001953**	0.1309	**0.001953**	**0.0039**
13	**0.007812**	**0.0019**	**0.005859**	**0.0019**	**0.001953**	**0.0039**	**0.001953**	**0.0019**	**0.001953**	**0.0019**	**0.001953**	**0.0019**
14	(-) 1	(-) 0.375	**0.007812**	0.1934	**0.009766**	**0.0039**	**0.001953**	0.1934	**0.001953**	**0.0058**	**0.001953**	**0.0039**

The statistically significant values (p-value $\leq \alpha/2$) are in boldface. The symbol "-" means that SMOTI performs worse than M5' or RETIS. All statistically significant values are favorable to SMOTI.

residuals whose sums are zero and, thus, the lines must pass through the origin.

The tree built by SMOTI on a cross-validated training set of 162 cases is shown in Fig. 3b. It well approximates the underlying model by discovering both global and local effects. The tree found by M5′ (see Fig. 3c) is less accurate on the validation set of the remaining 18 cases and is not easily interpretable, especially because of the smoothing process adopted by the system to compensate for the sharp discontinuities that occur between linear models at adjacent regions [23].

The clear superiority of SMOTI on these data sets should not be surprising since neither M5′ nor RETIS have been designed to discover both global and local effects of variables in the underlying data model. However, this has a computational cost. Fig. 4 plots the computation time of the three systems for the 90 artificial data sets. Naturally, M5′ is the most efficient because of its evaluation function, which is incoherent with respect to the model tree being built. RETIS has time performance comparable to SMOTI for small data sets (about 650 cases), while it becomes surprisingly faster than SMOTI for larger data sets. Coherently with our theoretical analysis, SMOTI shows a quadratic behavior, while RETIS does not. There are two possible explanations of RETIS efficiency. First, our theoretical analysis of RETIS computational complexity refers only to continuous variables since the case of discrete variables is undocumented. If RETIS used the same criterion applied in CART and M5′, then it would be more efficient than SMOTI, but its evaluation function could no longer be considered coherent with the models at the leaves. Second, an undocumented stopping criterion prevented the system from generating large model trees in the experiments.[5] Finally, we observe that several optimizations can still be implemented in SMOTI. In particular, the recursive least squares algorithm proposed by Torgo [22] fits very well SMOTI learning strategy.

4.2 Experiments on Benchmarks for Regression and Model Trees

SMOTI was also tested on 14 data sets (see Table 3) taken from either the UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/MLRepository.html) or the site of the system HTL (http://www.niaad.liacc.up.pt/~ltorgo/Regression/DataSets.html) or the site of WEKA (http://www.cs.waikato.ac.nz/ml/weka/). They have a continuous variable to be predicted and have been used as benchmarks in related studies on regression trees and model trees.

In all experimental results reported in this section, the thresholds for the stopping criteria are set at the same values used in the experiments on artificial data sets, except for the coefficient of determination that is set at 0.9. Experimental results are reported in Table 4, where SMOTI is compared to M5′ and RETIS on the basis of the average MSE. As in the previous experimentation, differences are considered statistically significant when the p-value is less than or equal to $\alpha/2$.

The comparison with RETIS is clearly in favor of SMOTI. Unfortunately, not all experimental results could be collected for RETIS because of two limitations of the system on the maximum number of attributes and on the maximum number of distinct values for discrete attributes.

[5] The system often outputs the message "Too many nodes. Making a leaf."

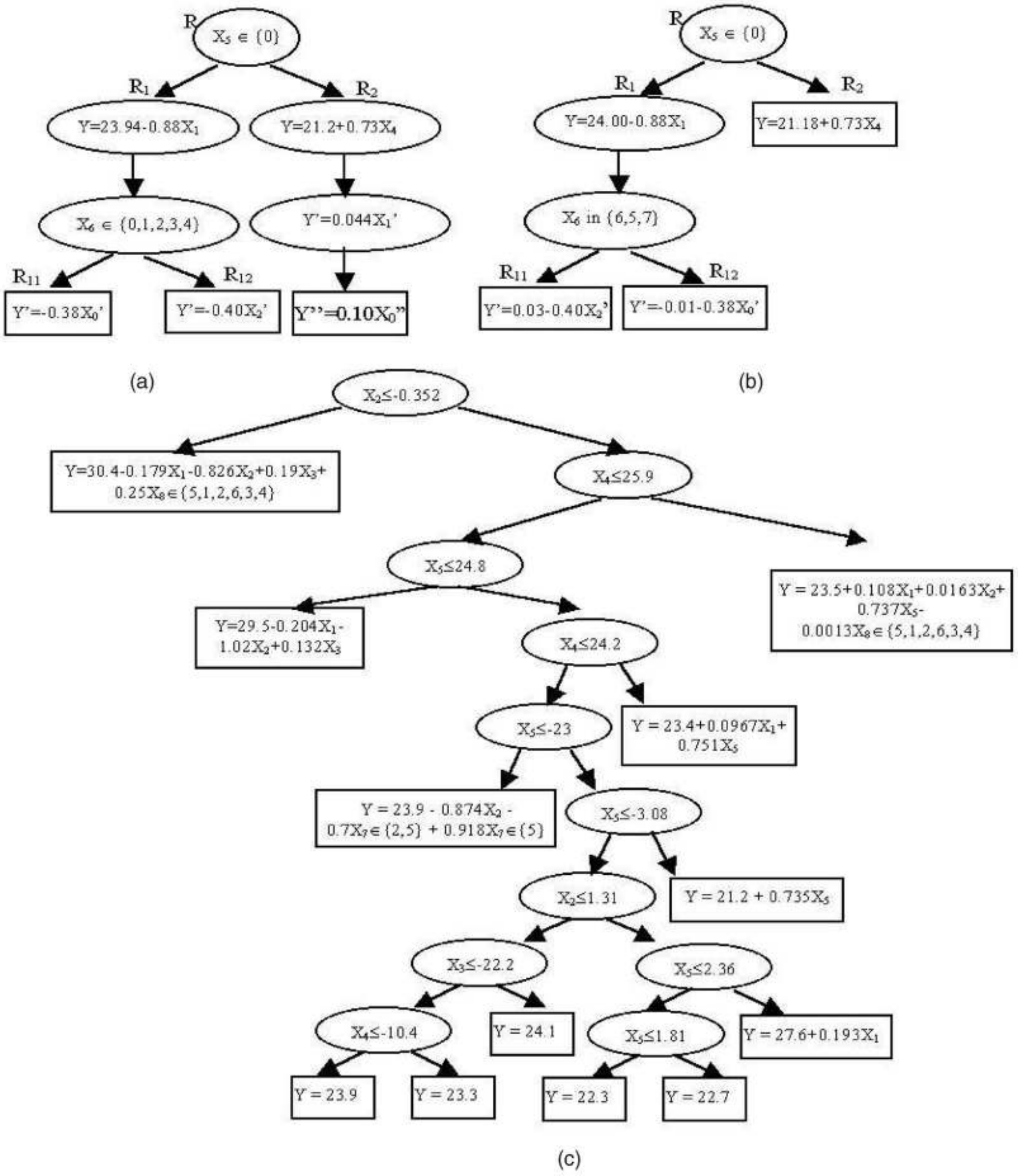


Fig. 3. (a) A theoretical model tree of depth 4 used in the experiments, (b) the model tree induced by SMOTI from one of the cross-validated training sets, and (c) the corresponding model tree built by M5' for the same data.

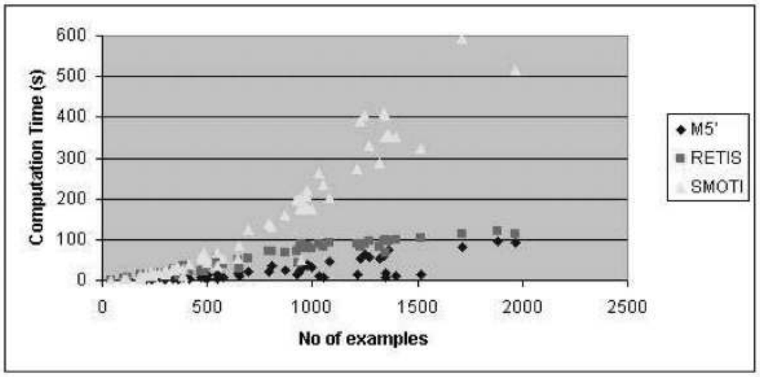
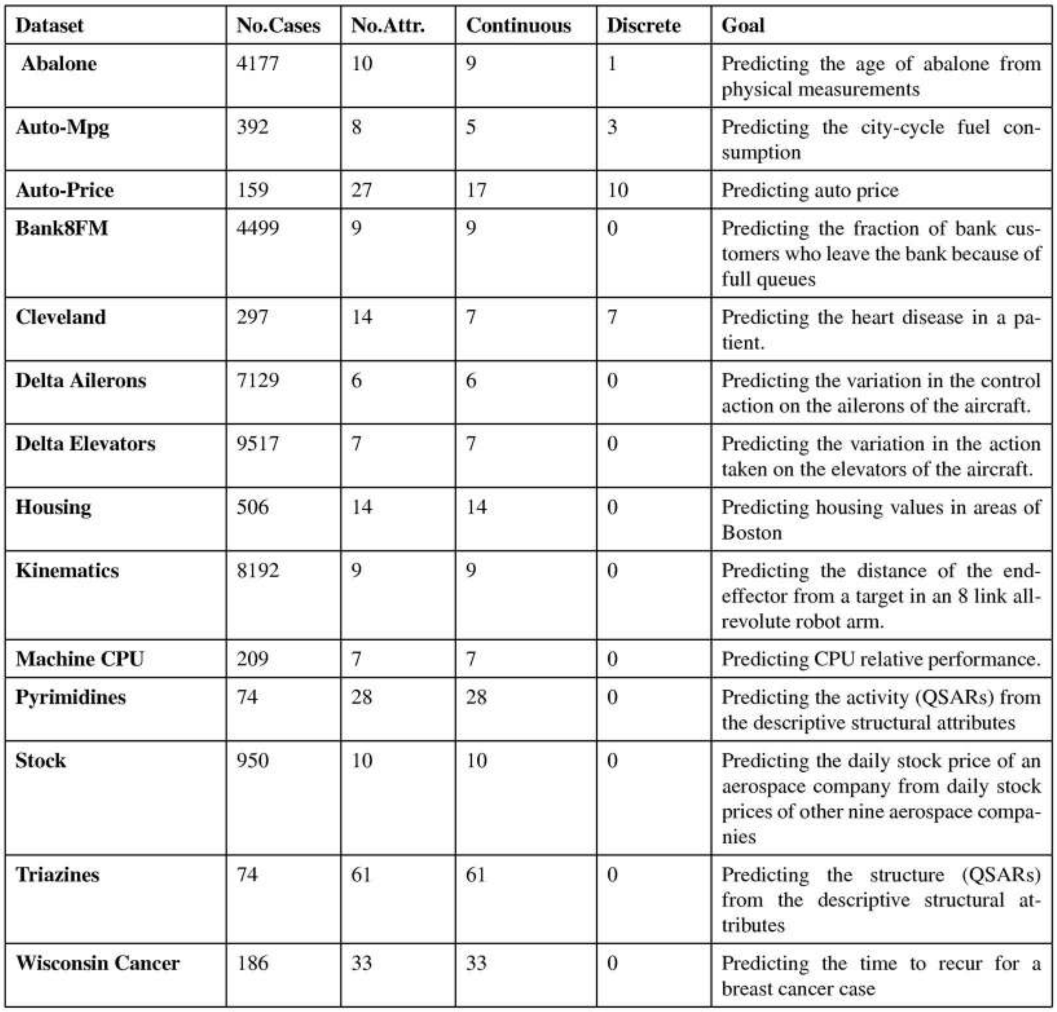


Fig. 4. Running time on artificial data sets. Experiments are performed on a PentiumIII PC-366MHz running Windows 98.

Differently from artificially generated data, SMOTI does not exhibit an irrefutable superiority with respect to M5', although results are still good. A deeper analysis of the experimental results evidenced that, for some training sets, the thresholds defined for the stopping criteria prevented SMOTI from growing model trees more accurate than those

TABLE 3
Data Sets Used in the Empirical Evaluation of SMOTI



Dataset	No.Cases	No.Attr.	Continuous	Discrete	Goal
Abalone	4177	10	9	1	Predicting the age of abalone from physical measurements
Auto-Mpg	392	8	5	3	Predicting the city-cycle fuel consumption
Auto-Price	159	27	17	10	Predicting auto price
Bank8FM	4499	9	9	0	Predicting the fraction of bank customers who leave the bank because of full queues
Cleveland	297	14	7	7	Predicting the heart disease in a patient.
Delta Ailerons	7129	6	6	0	Predicting the variation in the control action on the ailerons of the aircraft.
Delta Elevators	9517	7	7	0	Predicting the variation in the action taken on the elevators of the aircraft.
Housing	506	14	14	0	Predicting housing values in areas of Boston
Kinematics	8192	9	9	0	Predicting the distance of the end-effector from a target in an 8 link all-revolute robot arm.
Machine CPU	209	7	7	0	Predicting CPU relative performance.
Pyrimidines	74	28	28	0	Predicting the activity (QSARs) from the descriptive structural attributes
Stock	950	10	10	0	Predicting the daily stock price of an aerospace company from daily stock prices of other nine aerospace companies
Triazines	74	61	61	0	Predicting the structure (QSARs) from the descriptive structural attributes
Wisconsin Cancer	186	33	33	0	Predicting the time to recur for a breast cancer case

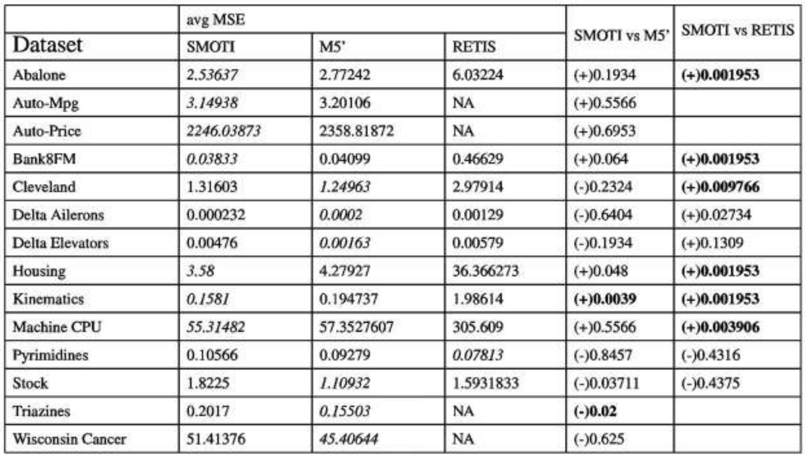
built by M5′. This problem cannot be straightforwardly solved by defining higher thresholds since that would lead to data overfitting problems. SMOTI can actually apply some postpruning strategy to reduce data overfitting; however, this aspect is beyond the scope of this paper.

The interesting aspect of this experimentation is that, for some data sets, SMOTI detected the presence of global effects that no previous study on model trees has revealed. In the following, we account for some of them, thus proving another desirable characteristic of the system, that of easy interpretability of the induced trees. The comparison is made with M5′ which outperforms RETIS.

Abalone. Abalones are marine crustaceans whose age can be determined by counting under the microscope the rings in the cross section of the shell. Other measurements, which are easier to obtain, can be used to predict the age. For all 10 cross-validated training sets, SMOTI builds a model tree with a regression node in the root. The straight-line regression selected at the root is almost invariant for all model trees and expresses a linear dependence between the number of rings (dependent variable) and the shucked weight (independent variable). This is a clear example of a global effect, which cannot be grasped by examining the nearly 350 leaves of the unpruned model tree induced by M5′ on the same data. Interestingly, the child of the root is always a splitting test on the whole weight or, more precisely, on the residuals of the whole weight once the effect of the shucked weight has been removed. As for the root, the threshold selected for this continuous split is almost the same for all 10 induced model trees. Unfortunately, this stability of the tree structure occurs only at the root and its child.

Auto-Mpg. The data concerns city-fuel consumption in miles per gallon. For all 10 cross-validated training sets, SMOTI builds a model tree with a discrete split test in the root. The split partitions the training cases in two subgroups, one whose *model year* is between 1970 and 1977 and the other whose *model year* is between 1978 and 1982. That can be easily explained with the measures for energy conservation prompted by the 1973 OPEC oil embargo. Indeed, in 1975, the US Government set new standards on fuel consumption for all Vehicles. These values, known as C.A.F.E. (Company Average Fuel Economy) standards, required that, by 1985, automakers doubled average new car fleet fuel efficiency. These standards came into force only in 1978 and model trees

TABLE 4
SMOTI versus M5' and RETIS: Results of the Wilcoxon Signed Rank Test on the Accuracy of the Induced Models



	avg MSE			SMOTI vs M5'	SMOTI vs RETIS
Dataset	SMOTI	M5'	RETIS		
Abalone	*2.53637*	2.77242	6.03224	(+)0.1934	**(+)0.001953**
Auto-Mpg	*3.14938*	3.20106	NA	(+)0.5566	
Auto-Price	*2246.03873*	2358.81872	NA	(+)0.6953	
Bank8FM	*0.03833*	0.04099	0.46629	(+)0.064	**(+)0.001953**
Cleveland	1.31603	*1.24963*	2.97914	(-)0.2324	**(+)0.009766**
Delta Ailerons	0.000232	*0.0002*	0.00129	(-)0.6404	(+)0.02734
Delta Elevators	0.00476	*0.00163*	0.00579	(-)0.1934	(+)0.1309
Housing	*3.58*	4.27927	36.366273	(+)0.048	**(+)0.001953**
Kinematics	*0.1581*	0.194737	1.98614	**(+)0.0039**	**(+)0.001953**
Machine CPU	*55.31482*	57.3527607	305.609	(+)0.5566	**(+)0.003906**
Pyrimidines	0.10566	0.09279	*0.07813*	(-)0.8457	(-)0.4316
Stock	1.8225	*1.10932*	1.5931833	(-)0.03711	(-)0.4375
Triazines	0.2017	*0.15503*	NA	**(-)0.02**	
Wisconsin Cancer	51.41376	*45.40644*	NA	(-)0.625	

The best average MSE is in italics. The statistically significant values (p-value $\leq \alpha/2$) are in boldface. The symbol "+" ("-") means that SMOTI performs better (worse) than M5' or RETIS. Most of statistically significant values are favorable to SMOTI. For RETIS, not all values are available since the system limits the number of attributes to 30 and the maximum number of distinct values for discrete attribute to 26.

induced by SMOTI capture this temporal watershed. Moreover, in the case *model year* between 1970 and 1977, SMOTI performs another discrete splitting test on the number of cylinders, while, in the case *model year* between 1978 and 1982, SMOTI introduces a regression step generally involving the variable *weight*. Also, this difference seems reasonable since it captures the different technologies (e.g., lightweight materials) adopted by automakers before and after the introduction of C.A.F.E. standards. Differently from SMOTI, model trees induced by M5' perform a first continuous splitting on the variable *displacement* (≤ 191 versus > 191) and a second splitting on the variable *horsepower* for both left and right child. A test on the variable *model year* appears only at lower levels.

Auto-Price. This data set consists of three types of entities: 1) the specification of an auto in terms of various characteristics, 2) its assigned insurance risk rating, and 3) its normalized depreciation as compared to other cars. Almost all induced trees have a regression node in the root which expresses a linear dependence between the price (dependent variable) and the normalized losses (independent variable). Interestingly, one of the findings of a recent study (February 2000) from the Highway Loss Data Institute (HLDI) is that "sports cars and luxury cars continue to have the worst claims losses among passenger cars for crash damage repairs under insurance collision coverages. Passenger vans have the best loss result." Therefore, the global effect of *normalized losses* is confirmed by independent studies. On the contrary, the continuous splitting test on the variable *curb weight* generally performed by M5' at the root of the induced model trees seems less intuitive.

Bank8FM. This data set is synthetically generated from a simulation of how bank customers choose their banks. The goal is predicting the fraction of bank customers who leave the bank because of long queues. The models induced by SMOTI from all 10 cross-validation sets are quite simple and are characterized by a chaining on six regression nodes starting from the root. In most of the trials, the model tree is actually a chaining of only regression nodes, thus revealing the multiple linear regression nature of the problem. As shown in Table 4, M5' also finds good predictive model trees, although they have about 400 leaves with as many regression models associated with them.

Cleveland. The domain is heart disease diagnosis and the data was collected from the Cleveland Clinic Foundation. The dependent variable refers to the presence of heart disease in a patient. It is an integer valued from 0 (no presence) to 4. The high average MSE measured for both SMOTI and M5' (> 1.2) shows the complexity of this prediction task. The tree models induced by SMOTI in almost all trials have a chaining of regression nodes involving the variables *ca* (number of major vessels (0-3) colored by flourosopy), *thalach* (maximum heart rate achieved), *age* (age in years), and *chol* (serum cholestoral in mg/dl). We actually do not know the criteria adopted by specialists to define the presence of heart disease in a patient, but it is likely that the final score was synthesized as a weighted linear combination of several factors with a global effect. Differently from SMOTI, M5' partitions by performing a test on the variable *thal*∈ {fixed defect; reversable defect} versus *thal*∈ {normal} or on the variable *cp* (chest pain type): asymptomatic versus {typical angina, atypical angina, nonanginal pain}. The error found by M5' on some leaves is null since M5' approximates the dependent variable with one of the admissible values (e.g., constants 0 or 1).

Delta Ailerons. The problem is that of grafting the skills of flying a F16 aircraft in a flight simulator from behavioral traces of a human expert. In this control problem, the independent variables describe the status of the airplane, while the goal is to predict the control action on the ailerons of the aircraft. It is not obvious which independent variables the human pilot uses; he may build more complex variables out of simple ones or may extract them from the landscape image. What we observe is that, in eight model trees induced through cross-validation, SMOTI selects regression nodes at the top four levels. Variables used in these nodes are the *roll-rate*, *diff-roll-rate*, *curr-roll*, and *pitch-rate*. This means that the only variable that seems to have a local effect is *curr-pitch*. Model trees induced by M5' are more complex and, therefore, more difficult to interpret.

Delta Elevators. This data set is also obtained from the task of controlling an F16 aircraft. The goal variable is related to an action taken on the elevators of the aircraft. As in the previous domain, for eight model trees induced through cross-validation, SMOTI selects regression nodes at the top five levels. Variables used in these nodes are the *diffclb*, *altitude*, *climb-rate*, *roll-rate*, and *diff-diffclb*. This means that the only variable that seems to have a local effect is *curr-roll*. Once again, the model trees induced by M5′ are more complex (generally more than 350 leaves).

Housing. This data set concerns housing values in the suburbs of Boston. The goal is that of predicting the median value of owner-occupied homes in 1,000s. By treating the independent variable *chas* (an indicator variable equal to 1 if a tract bounds the Charles River, 0 otherwise) as continuous, SMOTI generally creates a model tree with the regression step *medv* = 22.09 + 5.6 *chas* in the root. Surprisingly, model trees induced by M5′ almost totally neglect this indicator variable.

Kinematics. This data set is synthetically generated from a realistic simulation of the forward kinematics of an eight link all-revolute robot arm. The goal is predicting the distance of the end-effector from a target, given the angular position of the joints [6]. Despite the claimed high nonlinearity of the data, SMOTI finds a model tree whose top seven nodes are all regression nodes involving the independent variables *theta3*, *theta5*, *theta6*, *theta1*, *theta8*, *theta2*, and *theta4*. After the introduction of the seven nodes, the algorithm starts partitioning the data set in many subregions, where linear dependencies on the remaining independent variable are considered. The simplicity of the model trees induced by SMOTI does not penalize their predictive accuracy since M5′ generates less accurate model trees with a thousand leaves.

Pyrimidines. The task consists of learning the Quantitative Structure Activity Relationships, in particular, the inhibition of dihydrofolate reductase by pyrimidines [11]. For this data set, both M5′ and SMOTI learn very simple model trees with few leaves. The model trees have almost the same predictive accuracy. Their main difference is that SMOTI detects the global effect of some variables. However, the limited training set size does not allow us to draw meaningful conclusions because of the instability of the tree structure built from the 10 cross-validated training sets.

Triazines. As for the Pyrimidines data set, the problem is to learn a model tree which predicts the activity from the descriptive structural attributes. The data and methodology are described in detail in [7], [10]. M5′ finds smaller and more accurate trees than those induced by SMOTI. Once again, the main difference is that SMOTI detects the global effect of some variables, but the limited training set size does not allow us to draw some conclusions on the tree structure.

5 Conclusions

TDIMT methods generally grow a tree structure in two phases. In the first splitting phase, leaf nodes are expanded and associated with split tests. In the second predictive phase, leaf nodes are labeled with a multiple linear model. One drawback with this tree-building strategy is that the choice of the split tests is often made independently of the type of model associated to the leaves. This could result in a model tree that does not capture the underlying data model, even in very simple cases that can be perfectly represented by a model tree. To overcome this problem, one of the TDIMT methods reported in the literature merges the two phases and chooses the best split test on the basis of the best multiple linear regression model associable to the leaves. Although correct, this approach considers only full regression models, while, in statistics, it is well-known that models based on subsets may give more precise results than will models based on more variables. This is due to the problem of collinearity. On the other hand, finding the best subset of variables while choosing the best split becomes too costly when applied to large data sets since it may require the computation of a high number of multiple linear regression models.

In this paper, we propose a new TDIMT method, SMOTI, which integrates the splitting phase and the predictive phase. Specifically, model trees generated by SMOTI include two types of nodes: regression nodes and splitting nodes. The former are associated with straight-line regression, while the latter are associated with split tests. Both types of nodes are considered at the same level during the tree construction process. This allows SMOTI to build the model tree stepwise and to overcome the computational problem of testing a large number of multiple linear regression models, while choosing the best split test with respect to the best multiple linear regression model at the leaves. In addition, this approach potentially solves the problem of modeling phenomena, where some variables have a global effect while others have only a local effect. Indeed, 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.

A comparison with two TDMTI systems, namely, M5′ and RETIS, has been reported for laboratory-sized data sets. It proves that SMOTI can induce more accurate model trees, when both global and local behaviors are mixed in the underlying model. However, the computation time of SMOTI is quadratic in the training set size, while it is linear for both M5′ and RETIS. The low-computation time of M5′ can be explained by the more efficient TDIMT strategy (i.e., the split test is chosen independently of the linear model associated with the leaves). Unfortunately, no clear justification can be given for RETIS efficiency, which is at variance with our computational complexity analysis.

The comparison has been extended to 14 benchmark data sets typically used to test regression tree induction algorithms. In this second experimentation, SMOTI clearly outperforms RETIS in accuracy, while it is not possible to draw statistically significant conclusions on the comparison with M5′. Model trees induced by SMOTI are generally simpler and can more easily be interpreted than those generated by M5′. The interesting aspect of this second experimentation is that, for some data sets, SMOTI detected the presence of global effects that no previous study on model trees has ever revealed.

The experimental results reported in this work are necessarily limited and do not include some important research tendencies. First, how model trees are induced by SMOTI compare to other approaches, such as neural networks. Obviously, model trees offer some advantages over neural networks, both computationally (no repetitive data feeding to converge toward a solution) and with respect to usability (the user is not forced to make guesses about the structure of the network to obtain accurate results). However, the neural networks can partition the feature space into irregular regions (e.g., ellipsoids), while model trees perform axis-parallel partitioning (and oblique partitioning when continuous split nodes are descendants of regression nodes). The hierarchical mixture-of-experts architecture presents some interesting similarities with SMOTI that will be empirically investigated in the near future [8]. The comparison can also be extended to support vector machines, which can be used for regression problems as well [15].

The second important research direction is the application of model trees induced by SMOTI to classification problems, as suggested by Frank et al. [5]. In this case, SMOTI can be used to predict class probabilities and, by learning multiple regression models instead of multiple linear models for each node, it would be possible to overcome the problem of building a separate tree for each class.

Similarly to many decision tree induction algorithms, SMOTI may generate model trees that overfit training data. Therefore, a third research direction is the a posteriori simplification of model trees with both regression nodes and splitting nodes. We plan to investigate simplification methods based on both pruning and grafting operators that require an independent pruning set. An extension of the MDL-based pruning strategies developed for regression trees [19] to the case of model trees with split and regression nodes is also under consideration since MDL-based pruning algorithms do not use an independent pruning set, which can be a problem when the data set is small.

References

[1] L. Breiman, J. Friedman, R. Olshen, and J. Stone, *Classification and Regression Tree.* Wadsworth and Brooks, 1984.

[2] P. Chaudhuri, M. Huang, W. Loh, and R. Yao, "Piecewise-Polynomial Regression Trees," *Statistica Sinica,* vol. 4, pp. 143-167, 1994.

[3] A. Dobra and J.E. Gehrke, "Secret: A Scalable Linear Regression Tree Algorithm," *Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* 2002.

[4] N. Draper and H. Smith, *Applied Regression Analysis.* John Wiley & Sons, 1982.

[5] E. Frank, Y. Wang, S. Inglis, G. Holmes, and I. Witten, "Using Model Trees for Classification," *Machine Learning,* vol. 32, pp. 63-76, 1998.

[6] Z. Ghahramani, D. Wolpert, and M. Jordan, "Generalization to Local Remapping of the Visuo-Motor Coordinate Transformation," *J. Neuroscience,* 1996.

[7] J. Hurst, R. King, and M. Sternberg, "Quantitative Structure-Activity Relationships by Neural Networks and Inductive Logic Programming. ii. The Inhibition of Dihydrofolate Reductase by Pyrimidines," *J. Computer-Aided Molecular Design,* vol. 8, pp. 421-432, 1994.

[8] M. Jordan and R. Jacobs, "Hierarchical Mixture of Experts and the EM Algorithms Neural Computation," *Neural Computation,* vol. 6, pp. 181-214, 1994.

[9] A. Karalic, "Linear Regression in Regression Tree Leaves," *Proc. Int'l School for Synthesis of Expert Knowledge,* pp. 151-163, 1992.

[10] R. King, J. Hurst, and M. Sternberg, "A Comparison of Artificial Intelligence Methods for Modelling Qsars," *Applied Artificial Intelligence,* 1994.

[11] R. King, R.L.S. Muggleton, and M. Sternberg, "Drug Design by Machine Learning: The Use of Inductive Logic Programming to Model the Structure-Activity Relationship of Trimephoprim Analogues Binding to Dihydrofolate Reductase," *Proc. Nat'l Academy of Sciences,* vol. 89, pp. 11322-11326, 1992.

[12] W. Loh, "Regression Trees with Unbiased Variable Selection and Interaction Detection," *Statistica Sinica,* vol. 12, pp. 361-386, 2002.

[13] D. Lubinsky, "Tree Structured Interpretable Regression," *Learning from Data,* D. Fisher and H. Lenz, eds., vol. 112, pp. 387-398, 1994.

[14] D. Malerba, A. Appice, M. Ceci, and M. Monopoli, "Trading-Off Local versus Global Effects of Regression Nodes in Model Trees," *Proc. Foundations of Intelligent Systems, 13th Int'l Symp.,* H.-S. Hacid, Z. Ras, D. Zighed, and Y. Kodratoff, eds., pp. 393-402, 2002.

[15] O.L. Mangasarian and D.R. Musicant, "Robust Linear and Support Vector Regression," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 9, pp. 950-955, Sept. 2000.

[16] M. Mehta, R. Agrawal, and J. Rissanen, "Sliq: A Fast Scalable Classifier for Data Mining," *Proc. Fifth Int'l Conf. Extending Database Technology,* pp. 18-32, 1996.

[17] M. Orkin and R. Drogin, *Vital Statistics.* New York: McGraw-Hill, 1990.

[18] J.R. Quinlan, "Learning with Continuous Classes," *Proc. Fifth Australian Joint Conf. Artificial Intelligence,* Adams and Sterling, eds., pp. 343-348, 1992.

[19] M. Robnik-Sikonja and I. Kononenko, "Pruning Regression Trees with MDL," *Proc. 13th European Conf. Artificial Intelligence,* H. Prade, ed., pp. 455-459, 1998.

[20] L. Torgo, "Functional Models for Regression Tree Leaves," *Proc. 14th Int'l Conf. Machine Learning,* D. Fisher, ed., pp. 385-393, 1997.

[21] L. Torgo, "Inductive Learning of Tree-Based Regression Models," PhD dissertation, Dept. of Computer Science, Faculty of Sciences, Univ. of Porto, Portugal, 1999.

[22] L. Torgo, "Computationally Efficient Linear Regression Trees," *Classification, Clustering and Data Analysis: Recent Advances and Applications (Proc. IFCS 2002),* K. Jajuga et al., eds., 2002.

[23] Y. Wang and I. Witten, "Inducing Model Trees for Continuous Classes," *Proc. Ninth European Conf. Machine Learning,* M. van Someren and G. Widmer, eds., pp. 128-137, 1997.

[24] S. Weisberg, *Applied Regression Analysis,* second ed. New York: Wiley, 1985.

Donato Malerba received the Laurea degree in computer science and, in 1992, he was an assistant specialist at the Institute of Computer Science, University of California, Irvine. He is an associate professor in the Department of Computer Science, University of Bari, Italy. His research activity mainly concerns machine learning and data mining, in particular, classification and model trees, numeric-symbolic methods for inductive inference, inductive-logic programming and relational data mining, spatial data mining, Web mining, and their applications. He has published more than 90 papers in international journals and conference proceedings. He is on the management board of KDNet (European Knowledge Discovery Network of Excellence) and AIIA (Italian Association for Artificial Intelligence) and is involved in many European and national projects on data mining, machine learning, and document processing. He has served on the program committee of many international conferences (ICML '96, '99; ISMIS '00, '02, '03; ECML '01, '02, '03, '04; PKDD '04; MLDM '01, '03, '05) and cochaired six international/national workshops and acted as guest-editor of three special issues of international journals (topics: machine learning in computer vision, mining official data, visual data mining). He is a member of the IEEE and the IEEE Computer Society.

Floriana Esposito received the Laurea degree in electronic physics from the University of Bari, Italy, in 1970. From 1974 to 1982, she was an assistant professor of system analysis and then (1984) an associate professor of computer science in the Computer Science Department of the University of Bari, Italy. Since 1994, she has been a full professor of computer science at the University of Bari and Dean of the Faculty of Computer Science from 1997 to 2002. She chairs the LACAM (Laboratory for Knowledge Acquisition and Machine Learning), the scientific laboratory she founded in 1986, in the Department of Computer Science. Her research activity started in the field of numerical models and statistical pattern recognition applied to the field of medical diagnosis. Then, her interests moved to the field of artificial intelligence and machine learning. The current research concerns similarity-based learning, the integration of numerical and symbolic methods in inductive learning, the logical and algebraic foundations of machine learning, multistrategy learning, computational models of incremental learning with the aim of refining and maintaining knowledge bases, revision of logical theories, knowledge discovery in data bases. Applications include document classification and understanding, content-based document retrieval, map interpretation, and semantic Web. She is a member of the IEEE and the IEEE Computer Society.

Michelangelo Ceci received the Laurea degree with honors in computer science from the University of Bari, Italy, in March 2001. He joined the Department of Computer Science at the University of Bari in 2001, where he is currently finishing his PhD thesis in data mining. He was a visiting scientist in the Department of Computer Science, University of Bristol, United Kingdom, in 2003-2004. His research interests are centered in knowledge discovery in databases, in particular, relational data mining and statistical approaches to inductive inference. Applications include document engineering, Web mining, map interpretation, and spatial data mining. He is involved in both European and national projects on intelligent document processing, data mining, and machine learning. He is a member of the Italian Association for Artificial Intelligence (AI*IA).

Annalisa Appice graduated in computer science from the University of Bari, Italy, in March 2001, discussing a Laurea thesis on data mining. Currently, she is a PhD student in the Department of Computer Science of the University of Bari and a visiting student in the Department of Computer Science, University of Bristol, United Kingdom. Her research activity mainly concerns machine learning and data mining, in particular, classification and model trees. She is also interested in data mining query languages, relational data mining, and data mining in spatial databases with applications to geographical information systems. She has published several papers on these topics in international journals, books, and refereed conferences. She is involved in European and national projects on machine learning, data mining, and symbolic data analysis. She is a member of the Italian Association for Artificial Intelligence (AI*IA).