Stepwise Induction of Multi-target Model Trees

Annalisa Appice[1] and Saso Džeroski[2]

[1] Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 - 70126 Bari - Italy
[2] Department of Knowledge Technologies, Jožef Stefan Institute
Jamova 39 - Ljubljana, Slovenia 1000
appice@di.uniba.it, Saso.Dzeroski@ijs.si

Abstract. Multi-target model trees are trees which predict the values of several target continuous variables simultaneously. Each leaf of such a tree contains several linear models, each predicting the value of a different target variable. We propose an algorithm for inducing such trees in a stepwise fashion. Experiments show that multi-target model trees are much smaller than the corresponding sets of single-target model trees and are induced much faster, while achieving comparable accuracies.

1 Introduction

Many problems encountered in ecological applications involve the prediction of several targets associated with a case. More formally, given a set of observed data $(\mathbf{x}, \mathbf{y}) \in \mathbf{X} \times \mathbf{Y}$, where $\mathbf{X}$ consists of m explanatory (or independent) variables X_i, the goal is to predict several target (or dependent) variables $Y_1, \ldots, Y_n$. The range of each Y_j can be either a finite set of unordered category labels for classification or a subset of real number $\Re$ for regression.

The problem of predicting several target variables simultaneously has been approached in the *predictive clustering* framework [1], where now methods exist to construct clusters of examples which are similar to each other and simultaneously associate a predictive model (classification or regression) with each constructed cluster. Several systems have been developed to induce decision and regression trees [1,9,5] or rules [8] within the predictive clustering framework, but to the best of our knowledge there is no attempt of inducing a *model tree* to predict the values of several continuous target variables simultaneously.

Model trees [3,10,6,7,4] are decision trees whose leaves contain linear regression models that predict the value of a single continuous target variable. In this paper, we address the task of inducing *multi-target* model trees that predict the values of several target continuous variables simultaneously. We propose an algorithm named MTSMOTI (*M*ulti *T*arget *S*tepwise *Mo*del *T*ree *I*nduction) that induces the multi-target trees in a stepwise fashion [2]. The tree is induced top-down by choosing at each step to either partition the training space (split nodes) or introduce a regression variable in the set of linear models to be associated with leaves (regression nodes).

J.N. Kok et al. (Eds.): ECML 2007, LNAI 4701, pp. 502–509, 2007.

The paper is organized as follows. The stepwise induction of multi-target model trees is presented in Section 2. Experimental results are reported in Section 3 and some conclusions are drawn in Section 4.

2 The Algorithm

Our system for the induction of multi-target model trees employs the basic stepwise construction of a regression model as it is implemented in SMOTI [4].

To explain the stepwise procedure, let us consider an example: suppose we are interested in analyzing a target variable Y in a region R described by two continuous explanatory variables X_1 and X_2 when R can be partitioned into two regions R_1 and R_2 and two linear regression models involving both X_1 and X_2 can be built independently for each region R_i. It may be found that the Y value is proportional to X_1 and this behavior is independent of any partitioning of R. In this case, the effect of X_1 on Y is *global*, since it can be reliably predicted for the whole region R. The initial regression model is approximated by regressing on X_1 for the whole region $R : \hat{Y} = \hat{a}_0 + \hat{b}_0 X_1$. The effect of another variable in the partially constructed regression model is introduced by eliminating the effect of X_1: we have to compute the regression model for the whole region R, that is, $\hat{X}_2 = \hat{a}_{20} + \hat{b}_{21} X_1$, as well as the residuals $X_2' = X_2 - \hat{X}_2$ and $Y' = Y - \hat{Y} = Y - (\hat{a}_0 + \hat{b}_0 X_1)$. The partitioning of R into R_1 and R_2 leads to building two independent regression models to capture the effect of the variable X_2 *locally* in the subregions R_1 and R_2, respectively. Obviously, a straight-line regression now involves the residual variables Y' and X_2', but can be automatically translated into a multiple linear function involving Y, X_1 and X_2.

This stepwise procedure corresponds to a tree structure with split nodes that produce binary partitions of the training data and regression nodes that perform straight-line regressions. Similarly to SMOTI, MTSMOTI induce such trees. However, MTSMOTI differs from SMOTI in several ways. First, MTSMOTI predicts several target variables simultaneously, assuming there is some (linear) dependence among target variables. Second, it resorts to a MAUVE [7]-based heuristic function to reduce the SMOTI time complexity of evaluating a node, yielding trees with better accuracy. Finally, it adopts some different stopping criteria and a post-pruning method. We discuss these topics below.

2.1 Model Tree Construction

The top-level description of the model tree construction performed by MTSMOTI is sketched in Algorithm 1.

A split node t on a variable X_i performs a binary test. If X_i is continuous, the split test is in the form $X_i \leq \alpha$ vs $X_i > \alpha$. Possible values of α are found by sorting the distinct values of X_i in the training sample falling in t, then identifying one threshold for each distinct value. If X_i is discrete, a discrete split partitions attribute values into two complementary sets, so that a binary tree is always built. To determine discrete split thresholds, we use the same criterion applied in CART. If $S_{X_i} = \{x_{i_1}, \ldots, x_{i_k}\}$ is the set of distinct values of X_i in t, S_{X_i} is sorted according to the sample mean of the target variable Y (or residual of Y) over all training cases falling in t, that is, $\bar{Y}_1, \ldots, \bar{Y}_k$. In the multi-target case, the set of distinct values of X_i is sorted according to the "average" of the sample means for each target variable Y_j from $\mathbf{Y}$. Since the range of different target variables may differ by several orders of magnitude, the sample means are scaled within the range $[0, 1]$. The scaled value of $\bar{Y}_{j_s}$ is $\bar{Y}_{j_{s \rightarrow [0,1]}} = \frac{|\bar{Y}_{j_s} - min_j|}{(max_j - min_j)}$, where $min_j = \min\limits_{s=1,\ldots,k}\{\bar{Y}_{j_s}\}$ and $max_j = \max\limits_{s=1,\ldots,k}\{\bar{Y}_{j_s}\}$.

Algorithm 1. MTSMOTI top-level description.

```
function build-MTSMOTI-tree($X, Y, R, E$) return T
$X$ → set of $m$ continuous ($X_C$) and discrete ($X_D$) explanatory variables
$Y$ → set of $n$ continuous target variables
$R$ → set of residuals of continuous variables; initially $R = X_C \cup Y$
$E$ → $\{(\boldsymbol{x}_j, \boldsymbol{y}_j) | j = 1 \ldots N\}$ a training sample
$T$ → a multi-target model tree with regression and split nodes
begin
$RegList$=regressionCandidates($X, Y, R, E$);
if stopping criteria then
    $t$ is the best regression node on $RegList$; $T$ =leaf($best_t$);
else
    $SplitList$=splitCandidates($X, Y, R, E$); $t$ is the best node on $RegList \cup SplitList$;
    if $t$ is a regression node on variable $X_i$ then
        $R'$ is a copy of $R$; $R_{X_i}$ is residual of $X_i$ in $R'$;
        for each $R_i \in R'$ do
            if $R_i$ represents either a target variable or a continuous explanatory variable
              not yet included in the current model then
                replace $R_i$ in $R'$ with its new residual by removing effect of $R_{X_i}$;
            end if
        end for
        $T'$ =build-MTSMOTI-tree($X, Y, R', E$); $T$ = tree with root in $t$ and child $T'$;
    end if
    if $t$ is a split node on variable $X_i$ then
        $T_L$ =build-MTSMOTI-tree($X, Y, R, \{(\boldsymbol{x}_j, \boldsymbol{y}_j) \in E|$ test in $t$ is true$\}$);
        $T_R$ =build-MTSMOTI-tree($X, Y, R, \{(\boldsymbol{x}_j, \boldsymbol{y}_j) \in E|$ test in $t$ is false$\}$);
        $T$ is the tree with root in $t$, left branch $T_L$, right branch $T_R$;
    end if
end if
end
```

A regression node performs a set of straight-line regressions on a continuous variable X_i, one for each target variable Y_j. Straight-line regressions in the subtree rooted in a regression node will involve residuals of both the target variables and the continuous explanatory variables not yet included in the model.

2.2 Split and Regression Node Evaluation

The choice of either a split test or a regression step at a node t is based on the evaluation measures $s(t, \boldsymbol{Y})$ and $r(t, \boldsymbol{Y})$, respectively.

Let t be a split on X_i then $s_j(t, Y_j)$ is computed as $s_j(t; Y_j) = \frac{N(t_L)}{N(t)} RE(t_L; Y_j) + \frac{N(t_R)}{N(t))} RE(t_R; Y_j)$, 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 $RE(t_L)$ $(RE(t_R))$ is the resubstitution error of the left (right) child. The resubstitution error is computed as $RE(t; Y_j) = \sqrt{\frac{1}{N(t)} \sum_{i=1}^{N(t)} (y_{j_i} - \hat{y}_{j_i})^2}$. For the left(right) child of a split t, the estimate $\hat{y}_j$ combines the straight-line regressions associated with regression nodes along the path from the root to t_L (t_R) with the straight-line regression on X_i computed on t_L (t_R). In case X_i is a discrete variable, straight-line regression on t_L (t_R) is replaced with the sample mean of Y_j (or residual of Y_j) values falling in t_L (t_R). This evaluation function is derived by MAUVE [7] as an alternative to consider no regression (M5')[10], simple regression on all continuous variables (SMOTI) or multiple regression on all continuous variables together (RETIS) [3]. The motivation in favor of the MAUVE measure is in its lower computational complexity. In fact, similarly to M5', MAUVE is linear in the number of variables, but the MAUVE split evaluation avoids some pathological behaviors of M5' [7]. The evaluation of a regression step $Y_j = \alpha_j + \beta_j X_i$ at node t is based on the resubstitution error $RE(t; Y_j)$. In this way, the selection of the best regression step requires the computation of a straight-line regression with complexity linear by number of examples falling in t, for each of the m target variables. Measures obtained at t for separate target variables are scaled to the interval $[0, 1]$ and combined as $s(t, \boldsymbol{Y}) = \frac{1}{n} \sum_{j=1}^{n} s_{\rightarrow[0,1]}(t; Y_j)$ $\left(r(t, \boldsymbol{Y}) = \frac{1}{n} \sum_{j=1}^{n} RE_{\rightarrow[0,1]}(t; Y_j)\right)$. The most promising split (regression) minimizes the evaluation measure s (r) on the set of split (regression) candidates.

As pointed in [4], a regression step on X_i would result in values of $r(t; Y_j)$ less than or equal to values of $s(t; Y)$ for some split test involving X_i. Hence, the split selection criterion in MTSMOTI is improved to consider the special case of identical regression models associated with both children (left and right): a useless split is replaced with a regression candidate. To check for this case, MTSMOTI compares pairs of lines associated with the children according to a statistical test for coincident regression lines [11] with linear time complexity.

2.3 Stopping Criteria

Three different stopping criteria are implemented. The first uses the partial F-test to evaluate the actual contribution provided by a new explanatory variable to the model [2]. The F-test is performed separately for each target variable. Hence, stopping can operate at different tree depth for different target variables. The second requires the number of examples in each node to be greater than a minimum value. The third stops the induction process when all continuous explanatory 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.

2.4 Pruning

MTSMOTI adopts a pruning procedure to determine which nodes of the tree should be taken as leaves and compute the set of linear models for each interior node of the un-pruned tree. Linear models built in a stepwise fashion at each node are expanded by sequentially adding variables, one at a time, on the basis of the strength of the average resubstitution error. The models are built by using only the continuous variables tested or regressed in the subtree below this node. For each target variable, the contribution of an added term is immediately evaluated according to the F-test and eventually dropped whenever it is not statistically significant. Once a linear model is in place for an interior node, the tree is pruned back from the leaves so long as the expected estimated error decreases. The estimate of the expected error is the average of the resubstitution errors (scaled in the range [0,1]) on the training cases reaching that node for each target variable. To avoid the underestimation of the expected error on unseen cases, the average resubstitution error is multiplied by the factor $(N(t)-\nu(t))/(N(t)+\nu(t))$, where $N(t)$ is the number of training cases that reach t an $\nu(t)$ is the number of variables in the linear model associated with the node.

3 Experimental Results

The performance of MTSMOTI is evaluated on both single-target and multi-target datasets by 10-fold cross-validation. For each target variable Y_j, we estimate the basis of the average relative mean square error ($RRMSE(D,Y_j) = \frac{1}{10}\sum_{i=1}^{10}(\sqrt{\sum_{h=1}^{N(D_i)}(y_{j_h}-\widehat{y}_{j_h}(D/D_i))^2}/\sqrt{\sum_{h=1}^{N(D_i)}(y_{j_h}-\bar{y}_j(D_i))^2}\ )$), where $\widehat{y}_{j_h}(D/D_i)$ is the value predicted for the j-th target variable of the h-th testing case by the model tree induced on D/D_i and $\bar{y}_j$ is the mean value of y_j on D_i. RRMSE is averaged on separate target variables. The complexity of trees is evaluated on the basis of the number of leaves. All the multi-target datasets as well as the results for PC-Tree reported in this Section are provided by Bernard Ženko [8].

3.1 Single-Target Datasets

MTSMOTI is tested on single-target datasets taken from the UCI Machine Learning Repository (http://www.ics.uci.). MTSMOTI is run in two settings. In the former setting (SR), model trees are built in a stepwise fashion, while in the latter setting (S), model trees are built by partitioning the training sample and then associating leaves with multiple linear models by post-pruning the tree. MTSMOTI is compared with REGTREE, i.e., our implementation of a regression tree learner, SMOTI, M5', predictive clustering trees (PCT) and rules (PCR). SMOTI and M5' are run with default stopping thresholds. The pruning of M5' is enabled, but no smoothing is used. Results are reported in Table 1.

Several conclusions are drawn from these experimental results. First, model trees outperform regression trees in accuracy and size. Second, our implementation of the stepwise tree construction generally improves performance of SMOTI

Table 1. Single-target regression: comparison of the average RRMSE and average size

	RRMSE						Size					
Dataset	MT (SR)	MT (S)	REG TR.	SMO TI	M5	PCT	MT (SR)	MT (S)	REG TR.	SMO TI	M5	PCT
AutoHorse	0.38	0.43	0.47	0.49	0.35	0.41	3.8	4.2	6.8	6.6	2.4	23
AutoMpg	0.40	0.44	0.44	0.46	0.37	0.45	12.7	10.5	16.5	10	4.8	16
AutoPrice	0.48	0.46	0.60	0.53	0.40	0.49	7.2	4.5	10.2	6.6	7.4	9
Tumor	1.00	1.16	1.16	1.34	0.99	0.96	4.7	17	16.6	8	1.1	3
Cloud	0.53	0.68	0.82	0.70	0.52	0.58	3.2	4.2	6.9	5	2.4	9
CPU	0.17	0.34	0.65	0.80	0.20	0.33	8.3	3.9	13	7	3.1	12
Housing	0.45	0.47	0.49	0.45	0.44	0.43	4.5	18.5	21.1	11	13.5	32
Quake	0.99	1.12	1.03	2.38	1.01	0.99	2.3	29	28.8	23	3.3	3
Sensory	0.93	0.93	0.93	1.00	0.96	0.94	16.5	16.5	16.5	12	4.7	7
Servo	0.60	0.60	0.60	0.45	0.43	0.42	7	7	7	7	5.6	11
Strike	0.94	0.91	0.96	1.88	1.24	0.98	3.9	18.7	18.7	12	6.5	12
Veteran	1.20	1.15	1.34	2.40	1.22	0.99	3.2	6.1	7.8	6	1	2

in accuracy. Third, the comparison between trees built in a stepwise fashion (SR) and trees in classical mode (S) show that trees with split and regression nodes achieve better (or at worst comparable) accuracy than trees with only split nodes. No general conclusion can be drawn on the tree size. Fourth, the comparison with M5' accuracy shows that MTSMOTI is sometime better, at worst comparable, to M5', but M5' typically builds smaller trees. In any case, MTSMOTI is able to detect the presence of global effects without significantly affecting accuracy. Finally, the comparison with predictive clustering trees shows that MTSMOTI does not exhibit an irrefutable superiority with respect to PC-TREE, although results are still good.

3.2 Multi-target Datasets

The multi-target datasets in this study are not public available. Only solar-flare (SOLARF) is available in the UCI Machine Learning Repository. A brief description of these datasets is reported in Table 2.

Table 2. Properties of multi-target datasets used in our study

Dataset	#Cases	#Explan. Var.	#Target Var.	Dataset	#Cases	#Explan. Var.	#Target Var.
EDM	154	16	3	SOLARF	323	10	3
MICROA	1944	142	3	WATERQ	1060	16	14
SIGMEAR	817	6	2	LANDSAT	60607	160	11
SIGMEAS	10368	11	2				

Table 3. Multi-target regression: comparison of the average RRMSE and average size

	RRMSE				Size			
Dataset	MTSMOTI	STSMOTI	MTREG TREE	PC T	MTSMOTI	STSMOTI	MTREG TREE	PC T
EDM	0.86	0.86	0.83	0.72	4	5	7	11
MICROA	0.78	0.60	0.87	1.01	18	47	17	50
SIGMEAR	0.71	0.91	1.15	0.85	3	8	11	7
SIGMEAS	0.03	0.03	0.03	0.03	23	27	49	166
SOLARF	1.02	1.06	1.02	1	13	30	13	2
WATERQ	0.96	0.97	0.98	0.96	12	92	13	5
LANDSAT	0.67	0.64	0.69	0.62	21	238	31.9	518

For each dataset, multi-target model trees (MTSMOTI) are first compared with the set of single-target model trees (STSMOTI), induced one for each target variable. The RRMSE is averaged over the target variables. The tree size for STSMOTI is the sum of size for all separate trees. Secondly, multi-target model trees are compared with multi-target regression trees (MTREGTREE) as well as predictive clustering trees. Results are reported in Table 3.

Results show that multi-target model trees are much smaller than the set of single-target model trees, while achieving comparable (sometime better) accuracy. MICROA is the only dataset where the multi-target model tree performs significantly worse than the set of single-target trees (0.78 vs. 0.60). A deeper analysis reveals that the worst performance involves only the prediction of one target variable (Shannon biodiversity: 0.7 vs. 0.28), while the accuracy estimates are comparable for the remaining two target variables (mites: 0.85 vs 0.82 and springehrtails: 0.78 vs 0.72). This negative result suggests the absence of a "linear" dependence between the Shannon biodiversity and the variables mites and springertails. In any case, multi-target trees are always induced much faster than the set of single-target ones (18 vs 25 (EDM), 202 vs 412 (MICROA), 5 vs 9 (SIGMEAR), 69 vs 84 (SIGMEAS), <1 vs 2 (SOLARF), 718 vs 1272 (WATERQ) and 9214 vs 25372 (LANDSAT): running times are in secs.

The comparison between multi-target model trees and regression trees reveals that although model trees are typically smaller than regression trees, they achieve comparable (or sometime better) accuracy than corresponding the regression trees. MTSMOTI is capable of detecting the presence of a global effect of some explanatory variable on "all" of the target variables that no previous study on these datasets have revealed. In this way, regression nodes implicitly reveal the existence of some linear dependences among the target variables at different depth of the tree hierarchy. Finally, the comparison with predictive clustering trees confirms are sometime more accurate than model trees (EDM and SIGMEA-REAL), but clustering trees can be significantly more complex. Finally, clustering trees predict the same constant values (for each example covered by the same leaf), and they are not be able of capturing any linear pattern in the data.

4 Conclusions

In this work, we present MTSMOTI, a system that induces multi-target model trees and predict the values of several target variables simultaneously. Leaves of such a tree contain several linear models, each predicting the value of a different target variable. Multi-target model trees are built with two types of nodes: split nodes and regression nodes. Experiments on single-target datasets shows that MTSMOTI is competitive with respect to regression tree learners, SMOTI and M5', as well as predictive clustering trees. Experiments on multi-target datasets confirm that multi-target model trees are much smaller than the set of single-target model trees, while achieving comparable accuracies. In addition, they are induced much faster. As future work, we plan to combine decision trees and model trees to predict continuous and discrete target variables, simultaneously. A comparison to predictive clustering rules should be interesting. Finally, adding linear regression models to predictive clustering rules is worth to be explored.

References

1. Blockeel, H., Raedt, L.D., Ramon, J.: Top-down induction of clustering trees. In: Shavlik, J. (ed.) Proceedings of the 15th International Conference on Machine Learning, pp. 55–63. Morgan Kaufmann, San Francisco (1998)
2. Draper, N.R., Smith, H.: Applied regression analysis. John Wiley & Sons, Chichester (1982)
3. Karalic, A.: Linear regression in regression tree leaves. In: Proceedings of International School for Synthesis of Expert Knowledge, Slovenia, pp. 151–163 (1992)
4. Malerba, D., Esposito, F., Ceci, M., Appice, A.: Top down induction of model trees with regression and splitting nodes. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(5), 612–625 (2004)
5. Struyf, J., Dzeroski, S.: Constraint based induction of multi-objective regression trees. In: Bonchi, F., Boulicaut, J.-F. (eds.) Workshop on Knowledge Discovery in Inductive Databases, pp. 222–233 (2005)
6. Torgo, L.: Functional models for regression tree leaves. In: Fisher, D. (ed.) Proceedings of the 14th International Conference on Machine Learning, pp. 385–393. Morgan Kaufmann, San Francisco (1997)
7. Vens, C., Blockeel, H.: A simple regression based heuristic for learning model trees. Intelligent Data Analysis 10(3), 215–236 (2006)
8. Ženko, B.: Learning Predictive Clustering Rules. PhD thesis, Faculty of Computer and Information Science, University of Ljubljana, Slovenia (2007)
9. Suzuki, M.G.W., Choki, Y.: k-nearest neighbour classification of symbolic objects. In: De Raedt, L., Siebes, A. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 436–446. Springer, Heidelberg (2001)
10. Wang, Y., Witten, I.: Inducing model trees for continuous classes. In: van Someren, M., Widmer, G. (eds.) ECML 1997. LNCS, vol. 1224, pp. 128–137. Springer, Heidelberg (1997)
11. Weisberg, S.: Applied regression analysis, 2nd edn. Wiley, Chichester (1985)