A Study on End-Cut Preference in Least Squares Regression Trees

Luis Torgo

LIACC-FEP, University of Porto
R.Campo Alegre, 823
4150 PORTO, PORTUGAL
email: ltorgo@liacc.up.pt,
URL: http://www.liacc.up.pt/~ltorgo

Abstract. Regression trees are models developed to deal with multiple regression data analysis problems. These models fit constants to a set of axes-parallel partitions of the input space defined by the predictor variables. These partitions are described by a hierarchy of logical tests on the input variables of the problem. Several authors have remarked that the preference criteria used to select these tests have a clear preference for what is known as end-cut splits. These splits lead to branches with a few training cases, which is usually considered as counter-intuitive by the domain experts. In this paper we describe an empirical study of the effect of this end-cut preference on a large set of regression domains. The results of this study, carried out for the particular case of least squares regression trees, contradict the prior belief that these type of tests should be avoided. As a consequence of these results, we present a new method to handle these tests that we have empirically shown to have better predictive accuracy than the alternatives that are usually considered in tree-based models.

1 Introduction

Regression trees [6] handle multivariate regression methods obtaining models that have proven to be quite interpretable and with competitive predictive accuracy. Moreover, these models can be obtained with a computational efficiency that hardly has parallel in competitive approaches, turning these models into a good choice for a large variety of data mining problems where these features play a major role.

Regression trees are usually obtained using a least squares error criterion that guarantees certain mathematical simplifications [8, Sec. 3.2] that further enhance the computational efficiency of these models. This growth criterion assumes the use of averages in tree leaves, and can be seen as trying to find partitions that have minimal variance (i.e. squared error with respect to the average target value). The main drawback of this type of trees is the fact that the presence of a few outliers may distort both the average as well as having a strong influence in the choice of the best splits for the tree nodes. In effect, as we will see in this

paper, the presence of outliers[1] may lead to the choice of split tests that have a very small sub-set of cases in one of the branches. Although these splits are the best according to the least squares error criterion they are counter-intuitive to the user and as we will see may even degrade predictive performance on unseen data. Users find it hard to understand that trees have top level nodes with branches that are very specific. Most users expect that top level nodes "discriminate" among the most relevant groups of observations (*e.g.* the observations with very high value of the target variable and the others).

The work presented in this paper addresses the problem of allowing this type of splits in regression trees, which is known as the end-cut preference problem [2, p.313-317]. We study this type of splits and their effect on both predictive accuracy and interpretability of the models. We compare this to the alternative of avoiding this type of splits in the line of what was proposed by Breiman and colleagues [2]. Our extensive experimental comparison over 63 different regression problems shows that the differences in terms of predictive accuracy of both alternatives are quite often statistically significant. However, the overall number of significant differences does not show a clear winner, which contradicts prior belief on the effect of end-cut preference in tree-based regression models.

In this paper we propose an alternative method that allows end-cut preference only in lower levels of the trees. The motivation behind this method is to avoid these splits in top level nodes, which is counter-intuitive for the users, but at the same time use them in lower levels as a means to avoid their negative impact in the accuracy of trees using least squares error criteria. Our experimental comparisons show a clear advantage of this method in terms of predictive accuracy when compared to the two alternatives mentioned before.

In the next section we present a brief description of least squares regression trees methodology and of the end-cut preference problem. Section 3 presents an experimental comparison between the alternatives of allowing and not allowing end-cut splits. In Section 4 we describe our proposed approach to handle the end-cut preference problem, and present the results of comparing it to the other alternatives. Finally, in Section 5 we provide a deeper discussion of the study carried out in this paper.

2 Least Squares Regression Trees

A regression tree can be seen as a kind of additive regression model [4] of the form,

$$rt(x) = \sum_{i=1}^{l} k_i \times I(x \in D_i) \tag{1}$$

where $k_i's$ are constants; $I(.)$ is an indicator function returning 1 if its argument is true and 0 otherwise; and $D_i's$ are disjoint partitions of the training data D such that $\bigcup_{i=1}^{l} D_i = D$ and $\bigcap_{i=1}^{l} = \phi$.

[1] These may be "real" outliers or noisy observations.

These models are sometimes called piecewise constant regression models. Regression trees are constructed using a recursive partitioning (RP) algorithm. This algorithm builds a tree by recursively splitting the training sample into smaller subsets. The algorithm has three key issues:

- A way to select a split test (the splitting rule).
- A rule to determine when a tree node is terminal.
- A rule for assigning a model to each terminal node (leaf nodes).

Assuming the minimization of the least squares error it can be easily proven (*e.g.* [8]) that if one wants to use constant models in the leaves of the trees, the constant to use in each terminal node should be the average target variable of the cases falling in each leaf. Thus the error in a tree node can be defined as,

$$Err\left(t\right) = \frac{1}{n_t} \sum_{D_t} \left(y_i - \overline{y}_t\right)^2 \tag{2}$$

where D_t is the set of n_t training samples falling in node t; and $\overline{y}_t$ is the average target variable (Y) value of these cases.

The error of a regression tree can be defined as,

$$Err\left(T\right) = \sum_{l \in \widetilde{T}} P\left(l\right) \times Err\left(l\right) = \sum_{l \in \widetilde{T}} \frac{n_l}{n} \times \frac{1}{n_l} \sum_{D_l} \left(y_i - \overline{y}_l\right)^2 = \frac{1}{n} \sum_{l \in \widetilde{T}} \sum_{D_l} \left(y_i - \overline{y}_l\right)^2 \tag{3}$$

where $\widetilde{T}$ is the set of leaves of tree T; and $P(l)$ is the probability of a case falling in leaf l (which is estimated with the proportion of training cases falling in the leaf).

During tree growth, a split test s, divides the cases in node t into a set of partitions. The decrease in error of the tree resulting from this split can be measured by,

$$\Delta Err\left(s,t\right) = Err\left(t\right) - \sum_i \frac{n_i}{n} \times Err\left(t_i\right) \tag{4}$$

where $Err\left(t_i\right)$ is the error on the subset of cases of branch i of the split test s.

The use of this formula to evaluate each candidate split would involve several passes through the training data with the consequent computational costs when handling problems with a large number of variables and training cases. This would be particularly serious, in the case of continuous variables that are known to be the major computational bottleneck of growing tree-based models [3]. Fortunately, the use of the least squares error criterion, and the use of averages in the leaves, allow for further simplifications of the formulas described above. In effect, as proven in [8], for the usual setup of binary trees where each node has only two sub-branches, t_L and t_R, the best split test for a node is the test s that maximizes the expression,

$$\frac{S_L^2}{n_{t_L}} + \frac{S_R^2}{n_{t_R}} \tag{5}$$

where $S_L = \sum_{D_{t_L}} y_i$ and $S_R = \sum_{D_{t_R}} y_i$.

This expression means that one can find the best split for a continuous variable with just a single pass through the data, not being necessary to calculate averages and sums of squared differences to these averages. One should stress that this expression could only be derived due to the use of the least squares error criterion and of the use of averages in the leaves of the trees[2].

Breiman and colleagues [2] mention that since the work by Morgan and Messenger [5] it is known that the use of the least squares criteria tends to favor end-cut splits, *i.e.* splits in which one of the branches has a proportion of cases near to zero[3].

To better illustrate this problem we describe an example of this end-cut preference occurring in one of the data sets we will use our experiments, the *Machine*[4] domain. In this data set the best split test according to the error criterion of Equation 5 for the root node of the tree, is the test $MMAX \leq 48000$. This split divides the 209 training cases in two sub-branches, one having only 4 observations. This is a clear example of a end-cut split. Figure 1 helps to understand why this is the best split according to the criterion of Equation 5.

As it can be seen in Figure 1, there are 4 observations (upper right part of the figure) that have end-cut values in the variable MMAX, and at the same time outlier values in the target variable. These are the two characteristics that when appearing together lead to end-cut splits. Within this context, a candidate split that "isolates" these cases in a single branch is extremely valuable in terms of the least squares error criterion of Equation 5.

Allowing splits like $MMAX \leq 48000$ in the example above, may lead to trees that seem quite ad-hoc to users that have a minimal understanding of the domain, because they tend to expect that top level nodes show highly general relations and not very specific features of the domain. This is reinforced by the fact that on most large data sets, trees do tend to be too deep for a user to grasp all details, meaning that most users will only be able to capture top-level splits. As such, although no extensive experimental comparisons have been carried out till now[5], it has been taken for granted that end-cut splits are undesirable, and most existing tree-based systems (*e.g.* CART [2], THAID [5] or C4.5 [7]) have some mechanism for avoiding them. However, if the drawbacks in terms of user expectations are irrefutable, as we will see in Section 3 the drawbacks of end-cut splits in terms of predictive accuracy are not so clear at all in the case of least squares regression trees.

[2] In [8] a similar expression was developed for the least absolute deviation criterion with medians on the leaves of the trees.

[3] For a formal proof of end-cut preference see [2, p.313-317].

[4] Available for instance in http://www.liacc.up.pt/~ltorgo/Regression/DataSets.html.

[5] To the best of our knowledge.

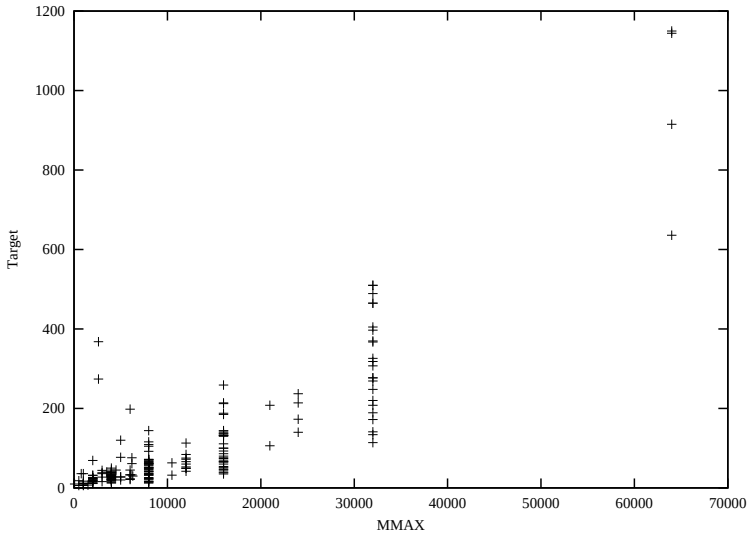

Fig. 1. An example of a end-cut preference problem.

3 An Experimental Analysis of the End-Cut Preference

In this section we carry out an experimental study of the consequences of end-cut splits. Namely, we compare the hypothesis of allowing this type of splits, and the alternative of using some form of control to avoid them.

In this experimental comparison we have used 63 different regression data sets. Their main characteristics (number of training cases, number of continuous variables, and number of nominal variables) are shown in Table 1.

Regarding the experimental methodology we have carried out 10 repetitions of 10-fold cross validation experiments, in the light of the recent findings by Bradford and Brodley [1] on the effect of instance-space partitions. Significance of observed differences were asserted through paired t-tests with 95 and 99 confidence levels.

The first set of experiments we report compares the following two types of least squares regression trees[6]. The first tree has no control over end-cut splits, thus allowing them at any stage of the tree growth procedure as long as they are better according to the criterion of Equation 5. The second type of trees does not allow splits[7] that lead to branches that have less cases then a minimum value

[6] Both are implemented in system RT (http://www.liacc.up.pt/~ltorgo/RT/), and they only differ in the way they handle end-cut splits. All other features are the same.

[7] Both on continuous as well as nominal variables.

Data Set	*Characteristics*	*Data Set*	*Characteristics*
Abalone (Ab)	4177; 7; 1	Elevators (El)	8752; 40; 0
Delta Elevators (DE)	9517; 6; 0	Ailerons(Ai)	7154; 40; 0
Kinematics (Ki)	8192; 8; 0	Telecomm (Te)	15000; 26; 0
ComputerA (CA)	8192; 22; 0	ComputerS (CS)	8192; 12; 0
Algal1 (A1)	200; 8; 3	Algal2 (A2)	200; 8; 3
Algal3 (A3)	200; 8; 3	Algal4 (A4)	200; 8; 3
Algal5 (A5)	200; 8; 3	Algal6 (A6)	200; 8; 3
Algal7 (A7)	200; 8; 3	Anastesia (An)	80; 6; 0
Auto-Mpg (AM)	398; 4; 3	Auto-Price (AP)	159; 14; 1
Bank8FM (B8)	4500; 8; 0	Bank32NH (B32)	4500; 32; 0
CloseNikkei (CN)	2000; 49; 1	CloseDow (CD)	2399; 49; 1
Chlorophyll (Chl)	72;4;1	House8L (H8)	22784; 8; 0
House 16H (H16)	22784; 16; 0	Diabetes (Di)	43; 2; 0
Pyrimidines (Py)	74; 24; 0	Triazines	186; 60; 0
FacultyD5001 (Fa)	197; 33; 0	Employment (Em)	368; 18; 0
ArtificialD2 (D2)	40768; 2; 0	Industry (In)	1555; 15; 0
Friedman Example (Fr)	40768; 10; 0	Housing (Ho)	506; 13; 0
Machine CPU (Ma)	209; 6; 0	Marketing (Mkt)	944; 1; 3
Artificial MV (MV)	25000; 7; 3	Puma8NH (P8)	4500; 8; 0
Puma32NM (P32)	4500; 32; 0	Servo	167; 0; 4
WiscoinBreastCancer (WBC)	194; 32; 0	CaliforniaHousing (CH)	20460; 8; 0
Additive (Ad)	30000; 10; 0	1KM (1KM)	710; 14; 3
Acceleration (Ac)	1732; 11; 3	CO2-emission (CO2)	1558; 19; 8
CW Drag (CW)	1449; 12; 2	Available Power (AP)	1802; 7; 8
Driving Noise (DN)	795; 22; 12	Fuel Town (FTw)	1764; 25; 12
Fuel Total (FTo)	1766; 25; 12	Fuel Country (FC)	1764; 25; 12
Maximal Torque (MT)	1802; 19; 13	Top Speed (TS)	1799; 17; 7
Maintenance Interval (MI)	1724; 6; 7	Heat (He)	7400; 8; 4
Steering Acceleration (SAc)	63500; 22; 1	Steering Angle (SAn)	63500; 22; 1
Steering Velocity (SV)	63500; 22; 1	Fluid Discharge (FD)	530; 26; 6
Fluid Swirl (FS)	530; 26; 6	China (Ch)	217; 9; 0
Delta Ailerons (DA)	7129; 5; 0		

Table 1. The Used Data Sets.

established by the user. In our experiments we have set this minimum value to 10 cases.

Table 2 shows the results of the comparison between the two alternatives in terms of Normalized Mean Squared Error (NMSE). Columns two and three show the data sets where we observed a statistically significant win of some method at different confidence levels, and the fourth column shows the cases where the observed differences were not statistically significant.

	99%	95%	Not significant
End-Cut Wins	**20** Ad,AP,B32,Chl,CO2,CW,D2, FS,FTo,FTw,He,Ma,MI,MT, An,Se,SAc,SAn,SV,TS	**2** FC,FD	**15** Ac,A1,A2,A3,A5,A6,A7, Ch,CN,Fa,DN,Ho,Te,AP,WBC
No End-Cut Wins	**17** 1KM,Ab,Ai,B8,CH,CD,CA, CS,DA,DE,El,Fr,H16,H8,Ki P32,P8	**2** In,Py	**7** A4,AP,Di,Mkt,MV,Em,Tr

Table 2. End-Cut versus No End-Cut in terms of NMSE.

The first thing to remark is that there is a statistically significant difference between the two approaches on 41 of the 63 data sets. This reinforces the importance of the question of how to handle end-cut splits. However, contrary to our prior expectations based on previous works (*e.g* [2]), we didn't observe a clear advantage of not using end-cut splits[8]. On the contrary, there is a slight advantage of the alternative allowing the use of end-cut splits at any stage of tree growth (the average NMSE over all data sets of this alternative is 0.4080, while not allowing end-cut splits leads to an average NMSE of 0.4140). The main conclusion to draw from these results is that they provide strong empirical evidence towards the need of a re-evaluation of the position regarding end-cut splits in the context of least squares regression trees.

Why should end-cut splits be beneficial in terms of predictive error? We believe the best answer to this question is related to the statistic used to measure the error. Least squares regression trees revolve around the use of averages and squared differences to these averages (*c.f.* Section 2). The use of averages as a statistic of centrality for a set of cases is known to suffer from the presence of outliers. By not allowing the use of end-cut splits that tend to isolate these outliers in a separate branch (*c.f.* Figure 1), every node will "suffer" the influence of these extreme values (if they exist). This will distort the averages, which may easily lead to larger errors as the predictions of the trees are obtained using the averages in the leaves. Going back to the example described in Section 2 with the *Machine* data set, if one does not allow the use of end-cut splits, instead of

[8] Still, we must say that the method proposed in [2] for controlling these splits is slightly different from the one used in our experiments.

immediately isolating the four outlier cases shown in Figure 1, they will end-up falling in a leaf that includes 10 other observations. This leaf has an average target variable value of 553.64, which will be the prediction of the tree for every test case falling in this leaf. However, 10 out of the 14 observations in this leaf, have a target value in the range [208..510]. Thus the distribution in this leaf is clearly being skewed by the outliers and this provides an idea of the risk of using this leaf to make predictions. The same conclusion can be reached by looking at the Mean Squared Errors[9] at the leaves of both trees. While the tree using end-cut splits has an average MSE over all leaves of 5132.2, the tree without end-cut splits has and average of 15206.4, again highlighting the effect of these outliers, that clearly increase the variance in the nodes. One should remark that this does not mean that the tree using end-cut splits is overfitting the data, as both trees went through the same post-pruning process that is supposed to eliminate this risk (moreover the experimental comparisons that were carried out show that this is not occurring, at least in a consistent way).

In resume, although clearly going against the intuition of users towards the generality of the tests in the trees, end-cut splits provide some accuracy gains in several data sets. This means that simply eliminating them can be dangerous if one is using least squares error criteria. We should stress that the same conclusions may not be valid if other error criteria were to be used such as least absolute deviations, or even the criteria used in classification trees, as these criteria do not suffer such effects of outliers.

As a consequence of the experimental results reported in Table 2 we propose a new form of dealing with end-cut splits that tries to fulfill the interpretability expectations of users that go against the use of end-cut splits, while not ignoring the advantages of these splits in terms of predictive accuracy. This new method is described in the next section.

4 A Compromising Proposal for Handling End-Cut Preference

The main idea behind the method we propose to deal with end-cut preference is the following. End-cut splits should not be allowed in top level nodes of the trees as they handle very specific (poorly represented in the training sample) areas of the regression input space, thus going against the interpretability requirements of most users. As such, our method will use mechanisms to avoid these splits in top level nodes of the trees, while allowing them in bottom nodes as a means to avoid the distorting effects that outliers have in the averages in the leaves.

In order to achieve these goals we propose a simple method consisting of not allowing end-cut splits unless the number of cases in the node drops below a certain user-definable threshold[10]. Moreover, as in the experiments of Section 3, a test is considered an end-cut split if one of its resulting branches has less than

[9] Larger values of MSE indicate that the values are more spread around the average.
[10] In the experiments we will report we have set this threshold to 100 cases.

a certain number of cases[11]. This means that nodes with a number of training cases between the first and second of these thresholds are allowed to consider end-cut splits. These are the bottom nodes of the trees[12].

Our hypothesis is that with this simple method we will obtain trees that are acceptable in terms of interpretability from the user perspective, but at the same time will outperform both alternatives considered in Section 3 in terms of predictive accuracy. With the purpose of testing this hypothesis we have carried out an experiment similar to the one reported in Section 3, but now comparing our proposed method with the two alternatives of allowing end-cut splits everywhere, and not allowing them at all. The results of comparing our method to the former alternative are shown in Table 3.

	99%	95%	Not significant
Our Method Wins	**16** Ab,Ad,Ai,CH,CD,CA,D2,DA, DE,El,H16,H8,In,Ki,Pu8,SV	**0**	**22** 1KM,A2,A3,A5,AP,B8,CS DN,FD,FTo,FTw,FC,He,MI Mkt,Em,Pu32,SAc,SAn,TS,Tr
End-Cut Wins	**2** A1,Fa	**0**	**23** Ac,A6,A7,AM,B32,Chl,Ch, CN,CO2,CW,Di,FS,Fr,Ho,Ma, MT,MV,Te,AP,Py,An,WBC,Se

Table 3. Our method compared to allowing end-cut splits in terms of NMSE.

This comparison clearly shows that there is no particular advantage in allowing end-cut splits everywhere, when compared to our proposal (with two single exceptions). Moreover, our proposal ensures that this type of splits will not appear in top level nodes of the trees[13], which fulfills the user's expectations in terms of interpretability of the models. In effect, going back to the *Machine* example, with our proposal we would not have a root node isolating the four outliers (as with the alternative of allowing end-cut splits), but they would still be isolated in lower levels of the tree[14].

What this comparison also shows is that our proposal can outperform the method of allowing end-cut splits in several data sets. It is interesting to observe that most of these 16 cases are included in the set of 17 significant losses of the alternative allowing end-cut splits shown in Table 2.

[11] We have used the value of 10 for this threshold.

[12] Unless the training sample is less than 100 cases, which is not the case in all but four of our 63 benchmark data sets (*c.f.* Table 1).

[13] Unless the data set is very small.

[14] Namely, the root node would consist of the split $MMAX \leq 28000$, which divides the 209 training cases in 182 and 27, respectively, and then the 27 cases (that include the 4 outliers) would be split with the end-cut test $MMAX \leq 48000$ (*c.f.* Figure 1 to understand what is being done with these splits).

The results of comparing our method to the alternative of not allowing end-cut splits are shown in Table 4.

	99%	95%	Not significant
Our Method Wins	**22** Ad,AP,B32,Chl,CD,CO2,CW,D2, FD,FS,FTo,FTw,He,Ma,MI,MT, An,Se,SAc,SAn,SV,TS	**2** FC,H8	**13** Ac,A2,A3,A5,A6,A7, DE,DN,H16,In,Mkt,AP,Tr
No End-Cut Wins	**10** 1KM,Ai,B8,CH,CA, CS,El,Fr,Ki,Pu32	**4** A1,Fa,Pu8,Py	**12** Ab,A4,AM,Ch,CN,DA, Di,Ho,MV,Em,Te,WBC

Table 4. Our method compared to not allowing end-cut splits in terms of NMSE.

Once again we observe a clear advantage of our proposal (24 significant wins), although there are still 14 data sets where not allowing end-cut splits seems to be preferable. However, comparing to the alternative of always allowing end-cut splits, which has clear disadvantages from the user interpretability perspective, our method clearly recovers some of the significant losses (*c.f.* Table 2). It is also interesting to remark that with the single exception of the *H8* data set, all 22 wins of the strategy using end-cut splits over the no-end-cuts approach, are included in the 24 wins of our proposal. This means that our method fulfills our objective of being able to take advantage of the gains in accuracy entailed by the use of end-cut splits, in spite of not using them in top levels of the trees.

In spite of the advantages of our proposal, there are also some drawbacks that should be considered. Namely, there is a tendency for producing larger trees (in terms of number of leaves) than with the other two alternatives that were considered in this study. This is reflected in the results shown in Table 5, that presents the comparison in terms of number of leaves of our proposal with the other two alternatives.

	No End-Cut Splits			All End-Cut Splits		
	99%	95%	Not significant	99%	95%	Not significant
Our Wins	**23**	**2**	**3**	**1**	**0**	**15**
Our Losses	**31**	**0**	**5**	**20**	**2**	**25**

Table 5. Tree size comparison of our method with the other two alternatives.

These results seem to contradict our goal of a method that produces trees more interpretable to the user than the trees obtained when allowing end-cut splits. Interpretability is known to be a quite subjective issue. Still, we claim that in spite of having a larger number of leaves (*c.f.* Table 5), the trees obtained with

our method are more comprehensible. As we have mentioned before, in most real-world large data sets, the trees obtained by this type of systems are too large for any user to be able to grasp all details. As such, we argue that only top-level nodes are in effect "understood" by the user. As our method does not allow end-cut splits in these top level nodes, we claim that this leads to trees that are more comprehensible to the user.

5 Discussion

The results of our empirical study on the effects of end-cut preference within least squares regression trees, lead us to the conclusion that there is no clear winner among the two standard alternatives of allowing or not allowing the use of end-cut splits. These results are somehow surprising given the usual position regarding the use of these splits. However, our study confirms the impact of the method used to handle these splits on the predictive accuracy of least squares regression trees. Our analysis of the reasons for the observed results indicates that this study should not be generalized over other types of trees (namely classification trees).

The method we have proposed to handle end-cut splits is based on the analysis of the requirements of users in terms of interpretability, and also on the results of our empirical study. By allowing end-cut splits only in lower levels of the trees, we have shown that it is possible to outperform the other two alternatives considered in the study in terms of predictive accuracy. Moreover, this method avoids end-cut splits in top level nodes which goes in favor of user expectations in terms of comprehensibility of the trees. However, our results also show that there is still some space for improvements in terms of predictive accuracy when compared to the alternative of not allowing end-cut splits. Future work, should be concentrated in trying to find not so ad-hoc methods of controlling these splits so as to avoid some of the still existing significant losses in terms of predictive accuracy. Moreover, the bad results in terms of tree size should also be considered for future improvements of our proposal.

6 Conclusions

We have described an empirical study of the effect of end-cut preference in the context of least squares regression trees. End-cut splits have always been seen as something to avoid in tree-based models. The main conclusion of our experimental study is that this assumption should be reconsidered if one wants to maximize the predictive accuracy of least squares regression trees. Our results show that allowing end-cut splits leads to statistically significant gains in predictive accuracy on 22 out of our 63 benchmark data sets. In spite of the disadvantages of end-cut splits in terms of the interpretability of the trees from the user perspective, these experimental results should not be disregarded.

We have described a new form of dealing with end-cut splits that tries to take into account our empirical observations. The simple method we have described

shows clear and statistically significant improvements in terms of predictive accuracy. Still, we have also observed that there is space for further improvements.

Future work should try to improve the method we have described, and also to carry out similar studies for tree-based models using different error criteria.

References

1. J. Bradford and C. Broadley. The effect of instance-space partition on significance. *Machine Learning*, 42(3):269–286, 2001.
2. L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth & Brooks/Cole Advanced Books & Software, 1984.
3. J. Catlett. *Megainduction: machine learning on very large databases*. PhD thesis, Basser Department of Computer Science, University of Sidney, 1991.
4. T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman & Hall, 1990.
5. J. Morgan and R. Messenger. Thaid: a sequential search program forthe analysis of nominal scale dependent variables. Technical report, Ann Arbor: Institute for Social Research, University of Michigan, 1973.
6. J. Morgan and J. Sonquist. Problems in the analysis of survey data, and a proposal. *Journal of American Statistics Society*, 58:415–434, 1963.
7. J. Quinlan. *C4.5: programs for machine learning*. Kluwer Academic Publishers, 1993.
8. L. Torgo. *Inductive Learning of Tree-based Regression Models*. PhD thesis, Faculty of Sciences, University of Porto, 1999.