Adapting Peepholing to Regression Trees

Luis Torgo[1] and Joana Marques[2]

[1] LIACC-FEP, University of Porto, R. de Ceuta, 118, 6., 4050-190 Porto, Portugal
ltorgo@liacc.up.pt,
WWW home page: http://www.liacc.up.pt/~ltorgo
[2] Aveiro Digital
jvcmarques@hotmail.com

Abstract. This paper presents an adaptation of the peepholing method to regression trees. Peepholing was described as a means to overcome the major computational bottleneck of growing classification trees by Catlett [3]. This method involves two major steps: shortlisting and blinkering. The former has the goal of eliminating some continuous variables from consideration when growing the tree, while the second tries to restrict the range of values of the remaining continuous variables that should be considered when searching for the best cut point split. Both are effective means of overcoming the most costly step of growing tree-based models: sorting the values of the continuous variables before selecting their best split. In this work we describe the adaptations that are necessary to use this method within regression trees. The major adaptations involve developing means to obtain biased estimates of the criterion used to select the best split of these models. We present some preliminary experiments that show the effectiveness of our proposal.

1 Introduction

Regression trees [5] handle multivariate regression problems 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. Nevertheless, the growth rate of databases creates problems even for these efficient methods. As such, techniques for reducing the computational requirements of growing regression trees are of key importance for handling extremely large problems.

Many strategies exist to try to overcome the problems of using a certain modelling technique on very large data sets. These can be classified in two broad categories: techniques that are independent of the modelling approach; and techniques that involve the optimization of model construction. In the former we can distinguish two major approaches: reducing the number of features; and reducing the number of cases. Regards the second category, the approaches are generally model-specific and basically try to address the computational bottlenecks of the respective algorithm for obtaining the models. The work presented in this paper

is of this type: we try to address the major computational bottleneck of growing regression trees.

One of the seminal works on handling large problems using tree-based models is the work by Catlett [3]. This author has thoroughly addressed the issue of the computational bottlenecks of growing classification trees, and presented several approaches to try to overcome them. One of the key contributions of Catlett's work was the identification of the major bottleneck of the process of growing a tree-based model: the selection of the best test on a continuous variable for any node of a tree. In this context, Catlett has presented a technique, *peepholing*, which specifically addresses the problem of reducing the computational complexity of this task. This technique consists of two major steps: *shortlisting* and *blinkering*. The first tries to eliminate some continuous variables from the set to be considered when searching for the best test of a node. The second tries to reduce the range of values to be considered when searching for the best cut point for a test in one of the variables not eliminated by the shortlisting step. Among these two steps, the former has the larger impact in terms of reducing the computational complexity of growing a tree according to Catlett.

Catlett has focused his work on classification trees. The peepholing method is strongly connected to the criterion used to select the best test of a node. The criteria used in classification trees are quite different from the usual criterion (least squares) used for growing regression trees. The goal of this work is to try to adapt the peepholing method to the growth of least squares regression trees. We identify the key issues that require modification and propose solutions in order to be able to use peepholing with regression trees.

The next section presents a brief overview of the theory behind the methods used to grow least squares regression trees. Section 3 describes the peepholing method presented by Catlett [3]. In Section 4 we describe our adaptation of this method to regression trees. The results of our preliminary experiments with this adaptation are shown in Section 5. Finally, the main conclusions of this work are given in Section 6.

2 Least Squares Regression Trees

Regression trees are usually obtained by using a least squares error criterion (e.g. [2]). A regression tree can be seen as a kind of additive regression model [4] of the form,

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

where $k_i's$ are constants; $I\left(.\right)$ 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 $\bigcap_{i=1}^{l} = \phi$.

These models are sometimes called piecewise constant regression models. Regression trees are constructed using a recursive partitioning (RP) algorithm

(e.g. [2]). 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 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. This is basically an estimate of the variance of the target variable obtained with the cases in node t.

The error of a regression tree can be defined as,

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

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

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.

For the usual binary regression trees, where each node has a left and a right branch, this reduces to,

$$\Delta Err\left(s,t\right) = Err\left(t\right) - \left(\frac{n_L}{n} \times Err\left(t_L\right) + \frac{n_R}{n} \times Err\left(t_R\right)\right) \tag{5}$$

For each iteration of the RP algorithm, assuming the stopping criteria are not yet met, we need to search for the best split test. This involves going through all variables of the problem and for each one finding the best test according to

the criterion of Equation (5). Moreover, for each variable we have to search for the best split, which for continuous variables involves sorting all values appearing in the data and evaluating all possible intermediate cut point using again Equation (5). Maximizing Equation (5) is equivalent to minimizing the second term, also known as pooled variance, as the first term, $Err(t)$ is constant for all candidate splits.

According to this process we can see that the computational complexity of growing regression trees is strongly dependent on the number of variables of the problem and also on the number of different values appearing on the data. Catlett has shown that continuous variables are particularly important given their large number of different values and the corresponding cost of the sorting operation. The peepholing method was designed to address these specific issues as we will see in the next section.

3 The Peepholing Method

The basic idea of the peepholing method proposed by Catlett [3] is to use a subsample (the *peephole*) to check whether some of the continuous variables, or some ranges within a continuous variable, can be removed from consideration for selecting the best split of a node without a significant loss in the overall accuracy, but with a significant gain in computation speed. Notice that, contrary to other related approaches (e.g. the sampling strategy proposed by Breiman et. al [2]), the final selection of the best test is carried out on all available data. The subsample is solely used for eliminating some candidates from this final selection process.

Peepholing is an iterative process where on each iteration we increase the size of the subsample (peephole) until certain criteria are met. For each peep size two main operations are carried out: *shortlisting* and *blinkering*.

Shortlisting consists of trying to eliminate some continuous variables from the list that will be used to select the best split of a node. The shortlist of a node starts with all continuous variables and if the estimates obtained using the current peephole size allow us to conclude that there is a very low probability that a certain variable would be a good candidate for the best split on the full sample, then that variable is eliminated from the shortlist. The variables eliminated by this process will not be considered on subsequent peepholes nor on the full sample best split selection.

Regarding blinkering it consists of maintaining a pair of numbers (the *blinkers*) for each variable in the shortlist. The interval spanned by the left and right blinkers is an estimate of the range where the best cut point for the respective variable may be contained. The process of blinkering tries to shorten this interval which brings gains in terms of computation time as we need to sort less values to find the best test of the variable.

Both steps of the peepholing process depend on obtaining reliable estimates of the gain of candidate tests using solely the data in the peepholes.

3.1 Shortlisting

The objective of shortlisting is to eliminate some of the continuous variables from the set of variables to be considered when selecting the best test for a node using all data. Catlett proposed a method based on a pair of biased estimates of the gain of a variable: the optimistic and the pessimistic estimates. These estimates are designed so that the gain assessed with the full data set is likely to lie between them. Once the estimates are obtained we can use them to eliminate all variables whose optimistic estimate falls below the greatest pessimistic estimate (GPE) of all variables in the current shortlist.

The method used for generating the biased estimates uses standard techniques from the estimation theory of statistics, namely from estimates of the mean of a population based on averages calculated with several samples with the same size.

The work of Catlett addresses classification trees grown using information gain [6] as the criterion for selecting the best split of a node. The information gain is not a function of the average of the values of examples. Instead its value is composed of several figures derived from averages, thus some adaptation of the general estimation theory was required. The basic building block of information gain is the expected information of a message. Assuming a binary class problem the expected information of the class is given by $-p \log p - n \log n$, where p is the probability of one of the classes and $n = 1 - p$. The values of p and n are obtained by estimating them with relative frequencies. These frequencies can be regarded as averages of a function $f(X) = \{1 \; if \; X = p, \; 0 \; if \; X = n\}$, over all possible messages X. The standard error of this estimate is computed using,

$$\sqrt{\frac{p(1-p)}{N-1}} \tag{6}$$

where N is the sample size.

By adding the standard error to the estimated frequency if p is less than 0.5, and subtracting it when it is greater, we get an optimistic estimate of p. Conversely, by subtracting the error when p is less than 0.5 and adding it when it is greater, we obtain a pessimistic estimate[3].

To calculate the information gain of a test on a variable V we need to calculate the gain resulting from its split of the examples. Assuming the standard binary splits the information content of a split is the weighed average of the information of the resulting sub-nodes entailed by the split,

$$InfGain(s, D_t) = Inf(D_t) - (p_L \times Inf(D_{t_L}) + p_R \times Inf(D_{t_R})) \tag{7}$$

where D_t is the set of examples in the node under consideration; p_L (p_R) is the probability of an example following the left (right) branch of the split test s; D_{t_L} (D_{t_R}) is the subset of D_t that fall in the left (right) branch of s; and $Inf()$ is the information content of a set of examples (given by $-p \log p - n \log n$).

[3] Obviously taking into account the limit 0 and 1 for probabilities.

The probabilities p_L and p_R are calculated using relative frequencies. Again we need to calculate optimistic and pessimistic estimates of these probabilities.

As the value of $Inf(D_t)$ is constant for all trial splits of the node t[4] , the best split s is the split that minimizes the weighed average of the information of the resulting branches, the estimate of the split. Catlett suggests obtaining a pessimistic estimate of the gain of a split by using the optimistic estimate of the split[5], and vice versa. In order to obtain an optimistic estimate of a split we combine the optimistic estimate of the most favorable branch with the pessimistic estimate of least favorable branch. The combination is carried out by giving more weight to the favorable branch by using the optimistic estimate of the probability of that branch, and obviously doing the inverse with the less favorable branch. The exact opposite process is done to obtain the pessimistic estimate of a split.

The above described process of shortlisting depends on obtaining optimistic and pessimistic estimates for two basic quantities: the information gain of a set of cases and the probability of a branch. These are the key issues that we need to address in order to adapt this process to regression trees.

3.2 Blinkering

The goal of blinkering is to restrict the range of values used for searching the best cut point of continuous variables. The idea is to choose an interval so that the best cut point is unlikely to fall outside of that range.

Catlett suggest the following heuristic to find the blinkers for each variable: eliminate from consideration the values whose information gain is less than half the maximum gain, or less than the average of the gains for all cut-points. These gains are calculated using the current peephole, and are updated for each new peep.

This step of peepholing requires no particular adaptation in order to use it in regression trees. Instead of the information gain of a test we use the least squares criterion of regression trees.

4 Adapting Peepholing to Regression Trees

There are two key issues that require modification in the peepholing method described in Section 3, if we want to apply it within regression trees. The most important is the adaptation of the criterion used to select the best split in a node. Catlett has described forms of obtaining pessimistic and optimistic estimates for the information gain, while in regression trees we need to provide similar facilities for the criterion used in these models (c.f. Equation (5)). The other

[4] This occurs because each candidate split of a node t only generates different partitionings of D_t, i.e. different values of D_{t_L} and D_{t_R}, but D_t in itself is the same set of cases.

[5] Because the largest the estimate of the split, the worse the value of the gain, according to Equation (7).

issue is to adapt the rule for eliminating some variables. This rule needs to be adapted because while information gain is being maximized, the variance used in regression trees is being minimized. This requires some adaptation in terms of what is a pessimistic (optimistic) estimate of the gain of a split.

As seen in Section 2, Equation (5), the best split is the split that minimizes the weighed average of the variance on the two sub-nodes resulting from the split (the pooled variance). Thus, provided we find means of obtaining a pessimistic and optimistic estimate of the variance of a node, we are ready to use a similar process as the one described in Section 3.1 for shortlisting. Regarding blinkering as we have mentioned before the method proposed by Catlett requires no adaptation in order to use it in regression trees.

A confidence interval for the sample variance of variable Y (S_Y^2) obtained with a sample of size N can be obtained by (e.g. [1]),

$$\left(\frac{(N-1)S_Y^2}{\chi^2_{\frac{\alpha}{2},N-1}}, \frac{(N-1)S_Y^2}{\chi^2_{1-\frac{\alpha}{2},N-1}} \right) \tag{8}$$

where $\chi^2_{\frac{\alpha}{2},N}$ is the value of the χ^2 distribution with a confidence level of α and N degrees of liberty.

Our proposal consists of using the smaller value of this interval as the optimistic estimate of the variance[6], while the largest value is used as the pessimistic estimate.

The quantity whose estimates we need to calculate to obtain the value of a test (c.f. Equations (5) and (2)) is the pooled variance,

$$var_{pool} = p_L \times var(D_{t_L}) + p_R \times var(D_{t_R}) \tag{9}$$

As soon as we have pessimistic and optimistic estimates of var_{pool} for a split s, we can calculate the respective estimates of the gain of this test as,

$$\begin{aligned} \Delta Err_{pess}(s,t) &= Err(t) - var_{pool_{pess}} \\ \Delta Err_{opt}(s,t) &= Err(t) - var_{pool_{opt}} \end{aligned} \tag{10}$$

This is different from the method described by Catlett due to the already mentioned fact that variance is something that we want to minimize on the tree.

We now need to define how to obtain the estimates for the pooled variance. Assuming the right branch is more favorable than the left (i.e. $var(D_{t_L}) > var(D_{t_R})$), the optimistic estimate of this quantity is given by,

$$var_{pool_{opt}} = p_{L_{pess}} \times var_{opt}(D_{t_L}) + p_{R_{opt}} \times var_{opt}(D_{t_R}) \tag{11}$$

The pessimistic and optimistic estimates of the probabilities of the branches can be calculated using,

[6] Recall, that contrary to information gain that we try to maximize, the variance is being minimized.

Table 1. The characteristics (number of cases, number of variables) of the data sets used in our experiments.

Data Set	Characteristics	Data Set	Characteristics
Abalone	4177; 9	Elevators	8752; 19
CaliforniaHousing	20460; 9	Ailerons	7154; 40
ComputerA	8192; 22	Puma32NM	4499; 33
House 16H	22784; 17	House 8L	22784; 9

$$p_{R_{opt}} = p_R + 3 \times SE$$
$$p_{L_{pess}} = p_L - 3 \times SE \tag{12}$$

where 3 corresponds roughly to a 99% confidence interval according to the normal distribution tables, and SE is the standard error of the probability estimate given by,

$$SE = \sqrt{\frac{p(1-p)}{N-1}}$$

where N is the sample size and p is the probability of the left[7] branch estimated by the relative frequency according to the current peephole.

On the other hand a pessimistic estimate of the pooled variance assuming the most favorable branch is still the right one, is given by

$$var_{pool_{pess}} = p_{L_{opt}} \times var_{pess}(D_{t_L}) + p_{R_{pess}} \times var_{pess}(D_{t_R}) \tag{13}$$

Equivalently, we could obtain similar formulas for the case where the left branch is more favorable.

Using these estimates $var_{pool_{pess}}$ and $var_{pool_{opt}}$ we can once again adapt Catlett's method, by discarding all continuous variables whose optimistic estimate is above the smallest pessimistic estimate (SPE) of all variables in the current shortlist.

5 Experimental Results

In this section we describe a series of preliminary experiments designed to assert the correctness of the adaptation we have described in Section 4. Namely, we have tried to assert the correctness of our pessimistic and optimistic estimates by checking whether they converge to the value obtained with the full training sample, as the size of the peephole increases. This is the key issue for using shortlisting to eliminate some variables from the tree growth process and thus strongly decreasing the computation time necessary to obtain these models.

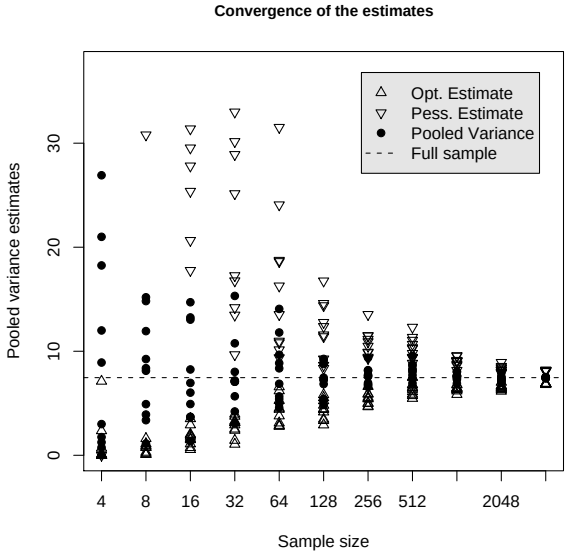

Fig. 1. The convergence of the estimates for the best root split in the Abalone data set ($Shell.Weight < 0.16775$).

We have used the data sets presented in Table 1 in our experiments.

With the goal of checking the convergence of our proposed pessimistic and optimistic estimates of the pooled variance, we have carried out the following experiment for each data set in Table 1:

1. Obtain the best split test using some regression tree program[8].
2. Divide the data set in two sub-samples according to this split.
3. Calculate the corresponding pooled variance using the full data set

$$var_{pool} = P_L \times var_L + P_R \times var_R$$

4. For increasing sizes of the peephole Do:
   - For several[9] random samples of the size under consideration Do:
     - Get a sample of the size under consideration
     - Calculate the pooled variance and its pessimistic and optimistic estimates using this sample.

[7] Or right, given that $p_L = 1 - p_R$.

[8] In our experiments we have used the function `rpart()` from the R environment [7] (http://www.r-project.org), which implements most of the ideas in the CART program [2].

[9] The graphs shown were obtained with 10 repetitions for each size.

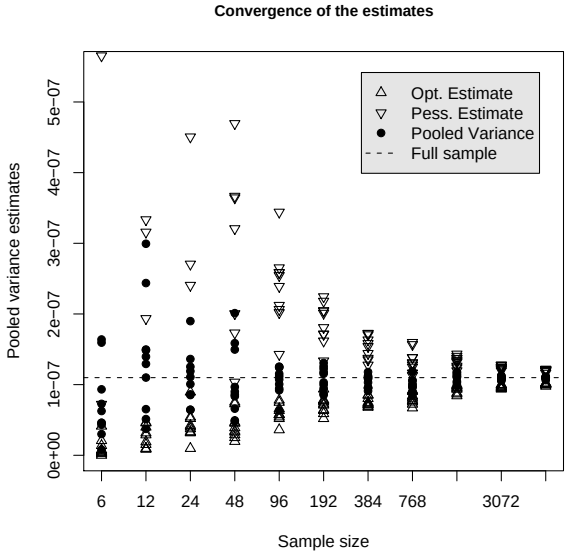

Fig. 2. The convergence of the estimates for the best root split in the Ailerons data set ($Goal < -13.5$).

Figures 1 and 2 show the results obtained for the best root splits for the data sets *Abalone* and *Ailerons*. The dashed line represents the value of the pooled variance for the best split, calculated using all training data. This is the value we want to approximate with our peephole estimates. The dots are the pooled variance estimates calculated with the peephole sample and the triangles present the respective pessimistic and optimistic estimates.

We can see in Figure 1 that for samples as small as 128 cases, which are roughly 3% of the full sample size we already obtain a quite acceptable convergence towards the pooled variance value obtained with the full sample. The same occurs in Figure 2 for sample sizes around 192, which correspond to 2.6% of the full sample size. Comparable results occur for all other data sets. These results clearly indicate that it is possible to obtain reliable estimates with small peephole sizes, which provides good indications towards the possibility of discarding some of the variables from the tree growth process. In effect, provided the variables have best splits which are reasonably different in terms of decrease in error according to Equation (5), we will be able to detect the "useless" variables if we have good estimates of their best splits using solely the information on the peephole. Obviously, in domains where all continuous variables are equally good in terms of their best split, the peepholing concept will not work as we will not be able to discard variables.

6 Conclusions

In this paper we have presented an adaptation of the work of Catlett [3] concerning the peepholing method. This strategy was created with the goal of overcoming the major computational bottleneck of growing classification trees. We have described the steps necessary to adapt this method to regression trees. Namely, we have shown how to obtain pessimistic and optimistic estimates of the variance of a node that are a key step in adapting Catlett's method.

Our preliminary experiments have shown that the estimates that we have developed clearly converge to the values obtained with the full sample, thus enabling their use in the shortlisting step of the peepholing method. The other step, blinkering, is directly applicable to regression trees thus not requiring any particular adaptation. As such, we have described all means to efficiently implement the peepholing method in any regression tree algorithm. In our future work we plan to carry out this implementation task for effectively using and testing the computational gains of peepholing in regression trees.

References

1. G. Bhattacharyya and R. Johnson. *Statistical Concepts and Methods*. John Wiley & Sons, 1977.
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 J. Sonquist. Problems in the analysis of survey data, and a proposal. *Journal of American Statistics Society*, 58:415–434, 1963.
6. J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1, 1986.
7. R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2004. ISBN 3-900051-07-0.