Extension of CART using multiple splits under order restrictions

R. Strobl, G. Salanti, K. Ulm

Institute for Medical statistics and Epidemiology
Klinikum rechts der Isar
Technical University of Munich
Munich, Germany
August 25, 2003

Abstract

CART was introduced by Breiman et al. (1984) as a classification tool. It divides the whole sample recursively in two subpopulations by finding the best possible split with respect to a optimisation criterion. This method, restricted up to date to binary splits, is extended in this paper for allowing also multiple splits. The main problem with this extension is related to the optimal number of splits and the location of the corresponding cutpoints. In order to reduce the computational effort and enhance parsimony, the reduced isotonic regression was used in order to solve this problem. The extended CART method was tested in a simulation study and was compared with the classical approach in an epidemiological study. In both studies the extended CART turned out to be a useful and reliable alternative.

1 Introduction

Data mining is the process of exploration and analysis of large quantities of data in order to discover meaningful patterns and rules. In many medical situations data mining tools are used in classification problems. One classic data mining tool seems to be notably efficient handling this kind of problems: the decision trees. Decision trees arise from a recursive partitioning of a sample.

One advantage of decision trees over traditional statistical methods like discriminant analyses or logistic regression is that it is non-parametric and it can handle interactions. This means that no assumptions about the distribution of the covariates are necessary. One disadvantage is, that the algorithm is restricted to binary splitting. Thus, one extension of the conventional CART algorithm is to allow non-binary splits. With the non-parametric isotonic regression, and the more sophisticated reduced isotonic regression, we have a method at hand, which can be used to detect cutpoints determining multiple splits in order to reduce the possible combinations of splits.

By combining the CART algorithm and isotonic regression a hybrid is formed which can shape trees more flexibly.

First the CART-algorithm is briefly described as a classification method. In addition we give an overview of the most important components of isotonic regression. This method has several advantages over parametric regression methods. Unlike them, it requires only the monotonicity assumption. Later we construct an algorithm which is able to build multiple trees by combining both methods. Furthermore we compare the results from this algorithm with the results from a conventional CART analysis in a simulation study. Finally, we apply the algorithm in an epidemiological study in investigating the influence of dust on developing chronic bronchitis in a Munich steel foundry.

For further informations we refer to Dannegger [4],[5]. The programme code can be found in Strobl [12].

2 Recursive Partitioning

In this section we give a brief introduction to the main components of tree construction. First, we introduce the method to grow a tree and to prune it back to a reasonable size. We follow mainly the ideas of Breiman et al. [1] and Zhang and Singer [13].

2.1 Growing Trees

CART can handle both type of explanatory variables: categorical and continuous ones. Even though there exists a variety of methods to handle continuous outcome (leading to regression trees), we focus in this paper on classification trees with binary response. Starting from a learning sample $\mathcal{L}$ where the explanatory variables x and the response y is known, CART divides $\mathcal{L}$ into disjunct subgroups with homogeneous outcomes in the explanatory variables and nearly identical results. To achieve this goal CART recursively partitions a sample into two subgroups (the daughter nodes) until no more meaningful split can be done that is when the sample contains only members of one class or no further split can is indicated. This splitting process is done with respect to one explanatory variable by means of "asking" each subject in the sample (in the beginning the $\mathcal{L}$) the same question: e.g. "Is the value higher than a certain cutpoint?". This successively partitioning of nodes is not at random. The algorithm rather chooses the best split among all possible splits. The number of possible splits is determined by the type of the covariate. For a ordered covariate the number of possible splits is simply $d-1$ with d the total number of *distinct* values of x. For a categorical one with k different categories there are $2^{k-1}-1$ possible splits. The algorithm picks the split which is best. The goodness of a split is determined by so-called *impurity functions* measuring the change in homogeneity between the parent node and the two daughter nodes.

Definition 2.1 (Impurity Measure) *The impurity measure $i(t)$ of node t is defined as*

$$i(t) = \phi(p(1|t), p(2|t), ..., p(J|t)).$$

ϕ is an impurity function on all J-tuples of $(p_1, p_2, ..., p_J)$ where p_i is the proportion of class i elements in the sample. $\sum_i p_i = 1$. $\phi(p_1, p_2, ..., p_J)$ must have the following properties:

1. *ϕ achieves its maximum for $(\frac{1}{J}, \frac{1}{J}, ..., \frac{1}{J})$. This means: $p_1 = p_2 = ... = p_J = \frac{1}{J}$*
2. *ϕ achieves its minimum 0 only for $(1, 0, 0, ..., 0)$ and any permutation of this*
3. *ϕ is a symmetric function of $(p_1, p_2, ..., p_J)$*

With an impurity function the goodness of a split is now defined as follows.

Definition 2.2 (Goodness of split) *The goodness of a split s is defined as the increase in homogeneity (or decrease in impurity)*

$$\Delta i(s, t) = i(t) - p_l \cdot i(t_l) - p_r \cdot i(t_r)$$

with p_l and p_r as the proportion of subjects falling into the left respectively right node.

In the following list we present the three most common impurity functions for binary response from which the *entropy* measure has the best reputation.

- Minimum Error: $i(p) = min(1, 1-p)$
- Entropy:$i(p) = -p \cdot log(p) - (1-p) \cdot log(1-p)$
- Gini: $i(p) = p \cdot (1-p)$

We can recursively find new two branches in the tree by splitting according to the highest $\Delta i(s,t)$. The partitioning procedure continues until all nodes (or leaves) contain only members of one class or are identical in their covariates. Due to the arbitrary character of the splits further down in the tree the growing process is often stopped earlier using stopping rules. The most common stopping rule is to limit the number of subjects in a node before it will be splitted again. A usual stopping rule would look for example like: if less than 5% of the total number of subjects are in a node the splitting stops.

2.2 Trees as classifiers

After a tree T is grown we still have to assign classes to each endnode (a node which is not further divided), so it can be used as a classifier. For this task we need a class assignment rule.

Definition 2.3 (Class Assignment Rule) *A class assignment rule assigns a class $j \in \{1, 2, ..., J\}$ to every terminal node $t \in \tilde{T}$. The class assigned is denoted by: $j(t)$.*

A simple and common class assignment is: $j^\star(t) = \max_j p(j|t)$. This rule assigns to a terminal node the class which occurs most often in it.

2.3 Assessment of Trees

There are several ways to assess the performance of a tree. All methods are based on a validation sample $\mathcal{V}$. The tree classifies $\mathcal{V}$. The number of misclassifications is used as a measure for the goodness of the tree $R^\star(T)$. The most important methods are:

- Resubstitution estimates: $R^{RE}(T)$
- Test sample estimates: $R^{TS}(T)$
- Bootstrap estimates: $R^{BS}(T)$
- Cross-validation estimates: $R^{CV}(T)$.

The most reliable results can be obtained using test sample estimates. In practical situations a test sample is often not available and we propose resample estimates to estimate the goodness of a tree.

2.4 Pruning Trees

A fully grown tree is usually to large to act as a good classifier. Therefore it is advisable to prune the tree back to reasonable size. With the estimators of tree performance (see section 2.3) it is possible to decide which tree is the best.

3 Isotonic Regression

Isotonic regression can be applied if we assume a monotonic relationship between the response and the covariable. Monotonic means with increasing x, y increases as well or the other way round. Without this assumption isotonic regression should not be used. On the other hand: to apply isotonic regression no assumption is needed and therefore the method is quite flexible. For further reading about this topic we refer to Barlow et al. [3],Robertson et al. [8] or Salanti and Ulm [10].

3.1 Pool Adjacent Violators Algorithm

In order to estimate isotonic regression we can use a very fast algorithm which is easy to implement and compute: the *Pool Adjacent Violator Algorithm* or short PAVA. Once we assume a monotonic relation (in medical settings a dose-response-relationship), we can use isotonic regression. The next list gives an overview how PAVA works.

For preparation we built blocks according to the following:

1. Order the response variable according to the covariable
2. Adjacent objects with the same value in response are grouped together forming a block. Therefore the remainder form blocks with only one object.
3. Weight the blocks with the number of objects in them.

Using these blocks we can run PAVA by merging adjacent blocks violating the isotonic assumption, the so-called violators.

1. Start with the first block and compare it to its successor.

2. If assumption is failed, than merge it and form a new block on level of the weighted average and adjust the weight of the new block.

3. Start again until no violators are left.

This simple algorithm yields the isotonic regression in form of a number of weighted blocks.

3.2 The Reduced Isotonic regression

The isotonic regression can be visualised as a step-function with a number of different level sets. Sometimes too many level sets remain and isotonic regression overfits the data. This means that not every level set arises from a significant change in the response. Therefore we would like to reduce the number of level sets to improve the parsimony of the model. We refer here to the works of Schell and Singh [11], Salanti [9] and Salanti and Ulm [10].

First all blocks are pooled with its precedessor (if no one exists than with its successor) containing less than a certain percentage of the total sample, for example 1%. In the next step all blocks which do not differ significantly (according to an exact Fisher test) from their neighbour are pooled. The significance level is determined in a simulation study in order to yield overall 5%.

In figure 1 we see the result of an isotonic regression and its reduced equivalent. It depicts the isotonic regression of $TIME$ against CBR in our test dataset. First, the whole dataset is ordered regarding $TIME$. Then the isotonic regression for CBR is calculated.

4 Modifications to allow non-binary splits

One limitation of CART is the binary splitting. One may argue that binary splitting is sufficient, because any non-binary split can be described by a sequence of binary splits. However, after each split the same variable competes with all the other variables. It may happen that this variable never will be used again and for this reason can not describe the non-binary relationship. Additionally in favour for non-binary splits can be the more simple structure of the trees. Several binary splits can hide a simple relation.

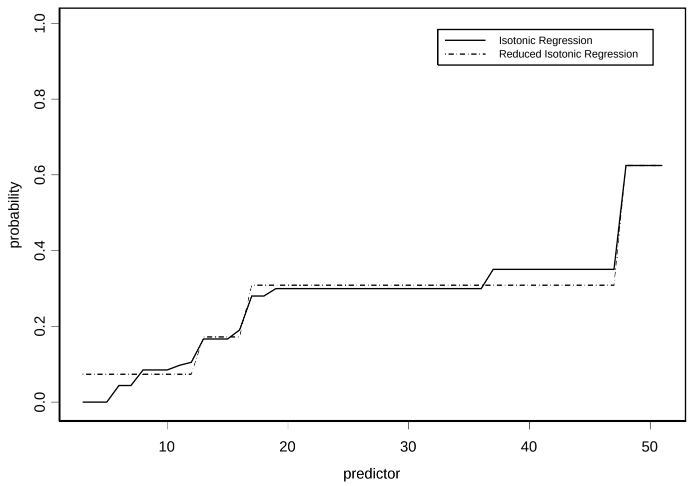

Figure 1: The isotonic regression and its reduced version

4.1 Implementation

To implement reduced isotonic regression in recursive partitioning some modifications in the CART algorithm should be done.

A main problem in splitting in more than two subnodes, is the fast increasing amount of possible splits and therefore the enormous number of comparisons that have to be done. We need a method for preselecting the cutpoints. At this point, the reduced isotonic regression comes into play. As we already mentioned, the result of reduced isotonic regression is a step function and yields several solution blocks. Each of these solution blocks does now act as a subnode.

Furthermore we have to modify the goodness of split criterion, because the number of subnodes should be taken into account. We suggest a likelihood ratio test and take the p-value as a measure of the goodness of the split, choosing the split with minimum p-value. The test has the following form:

$$T_{LR}^{S} = 2 \cdot \log \left(\frac{\prod_{s=1}^{S} L_s(\hat{\theta}_s)}{\prod_{s=1}^{S} L_s(\hat{\theta})} \right) \tag{1}$$

with $L_s(\hat{\theta}_s)$ as the likelihood of the subnode s using the estimators $\hat{\theta}_s$ for each subnode.

The test T_{LR}^S is asymptotically χ^2 distributed with $(S-1)\cdot(J-1)$ degrees of freedom with S as the number of subnodes and the J as the number of categories of the response. Independently of the type of the explanatory variable, even with a categorical one, this test can be applied and therefore all possible splits can be compared by means of the p-value. A process similar to the variable selection in multiple regression.

5 Simulation studies

In this section we investigate the properties of the tree algorithm with multiple splits in a simulation study. In this simulation study the underlying distribution consists of step functions, which can easily be described in a tree. We will have a closer look on the ability of the method to find the right number of cutpoints and their correct location. Therefore, we analyse the ability of the algorithm to find and use these steps as changepoints. We created 100 datasets with binary response each with size 1000. For each of the datasets a tree is grown using the reduced isotonic regression as changepoint detector. No stopping rules have been enforced.

5.1 Description of the Simulation

In the following x_1 and x_2 describe the distribution of the data. Both predictors are drawn from a rectangular distribution with a range of $[0, 100]$ and the response originates from a binomial distribution with probability p (see figure 2). In the diagram the labeling of the cutpoints always refers to the subnode on the left, beneath them. For example the first split divides the sample into three subgroups: on the left all cases with $x_1 < 33.33$ (#2), in the middle all cases with $33.33 \leq x_1 < 66.66$ (#3) and on the right all case with $66.66 \leq x_1$ (#4).

5.2 Results of the Simulation Study

All of the analysed trees use the variable x_1 as the first splitting variable. Furthermore 90% of the trees partition the root node into three subnodes, so perform a three-way split first. Of the remaining 10% is one half performing a binary split and the other half of them a four-way split.

For the trees which performed a three-way-split we give the density plots of the first and the second split in figure 3. The means are depicted in table 1.

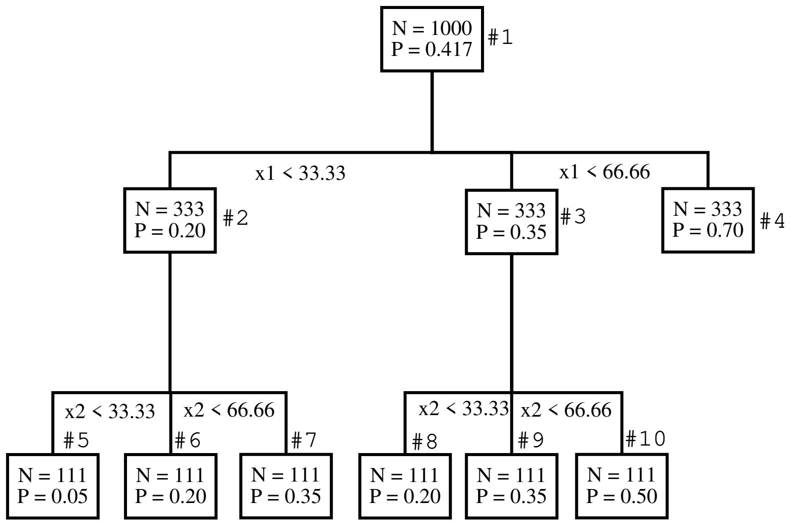

Figure 2: Description how data of the simulation study were generated.

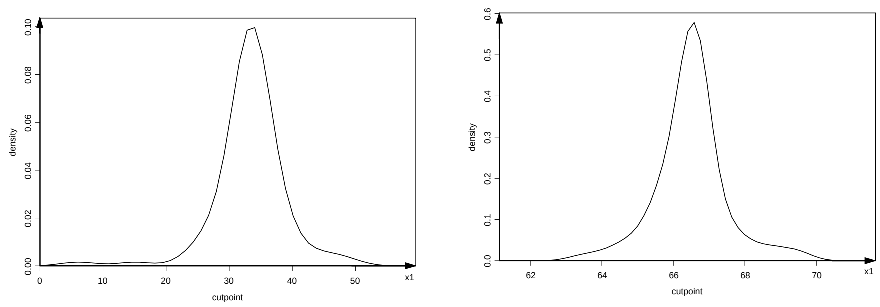

Figure 3: Distribution of the first (left figure) and second (right figure) cutpoint in the first split of the parent node

The trees preferring a binary split of the root node do all perform the split near 66.66 with a mean of 66.33. So, although they miss the first cutpoint, they at least detect the cutpoint leading to node 4 which identifies a very interesting group.

The trees finding three cutpoints do also all have a cutpoint of about 66.66. So, all trees do have a split leading to node 4. A summary of the first split can be seen in table 1.

Another interesting result is that 92% of the trees do not split node 4 further. The remainder 8% do find a single split in this node. The ability of the algorithm to stop at the right point seems to be well-trained.

If we go further down the tree and investigate the next layer of the tree we concentrate on the 90% performing the correct three-way split first. A first glance

Table 1: Summary of the first split (100 simulations)

First Split			
Variable Used	Number of Subnodes	Number of trees	Cutpoints (mean)
x_1	2	5	66.33
	3	90	33.33 and 66.50
	4	5	various

shows that none of them splits using variable x_1 again and apart from one tree, all of them find a split. The remainder 89 trees can be separated into two parts: one part contains the 56%, which find a binary split and the trees using a three-way split. In 95% of the cases the algorithm declares all the subnodes of node 2 as endnodes.

The first cutpoint of the trees using a binary split in node 2 does have a median of 35.96 and a mean of 40.53 and its density plot can be seen in figure 4. A summary of the split at node 2 can be found in table 2.

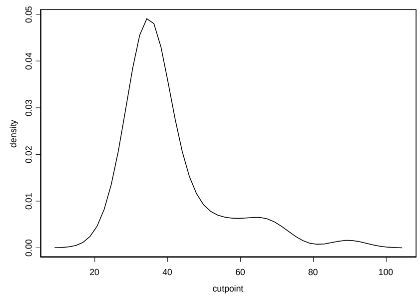

Figure 4: Distribution of the cutpoint value in the split at node 2 - only binary splits

The fact that the algorithm seems to prefer the cutpoint leading to endnode 5 and to merge endnode 6 and endnode 7 to one single endnode seems at first sight strange but can easily be explained if we bring back to mind the way the reducing process works. The reduction follows according to the p-value of an exact fisher test. It lies in the nature of the Fisher test, that it yields far lower p-values for a situation like the one between node 5 and 6 than for the one between 6 and 7.

The cutpoints for the 44% trees partitioning node 2 correct into 3 endnodes can be summarised by their means, namely: 33.45 and 68.36.

Finally we will have a closer look on the 90 trees starting with a three-way and

Table 2: Summary of the second split in node 2 for trees performing a three-way split first (90 simulations)

Split of node 2			
Variable Used	Number of Subnodes	Number of trees	Cutpoints (mean)
x_2	1	1	—
	2	50	40.53
	3	39	32.19

their ability of detecting node 8, 9 and 10. Here, only 18 prefer the correct three-way split while 72 of them find one cutpoint. The cutpoint of the binary split shows here, in contrary to the binary split of node 2, two peaks (see also figure 5: one around 33.33 and one around 66.66).

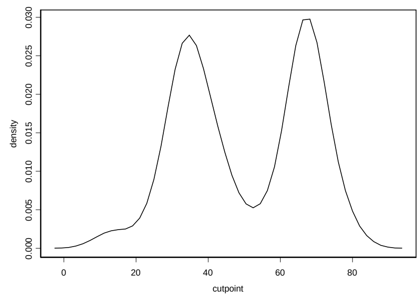

Figure 5: Distribution of the cutpoint value in the split at node 3 - only binary splits

The cutpoint of the models dividing node 3 into three subnodes are more or less all centred around the theoretical ideal with means of 33.96, respectively 79.06 (see also 3). As after the first split of node 2, most of the trees dispense with a further splitting of the generated subnodes.

Moreover, we will have a short look on the size of the trees. In table 4 we give an overview of the number of endnodes for all the trees.

The question arises if these endnodes are indeed similar to the endnodes in the tree in 2. In table 5 we give the number and names of the endnodes detected by the trees with 6 endnodes (the modus).

In all of them one endnode is similar to endnode 4. Only one of them misses endnode 5. Certainly due to the construction of a Fisher test in the reducing process, this endnode can easier be found. This might also be the reason, that

Table 3: Summary of the second split in node 3 for trees performing a three-way split first (90 simulations

Split of node 3			
Variable Used	Number of Subnodes	Number of trees	Cutpoints mean - median
x_2	2	72	50.28 (see figure 5)
	3	18	33.11 - 33.96 79.07 - 79.06

Table 4: Summary of the number of endnodes for all the trees

Endnode number	4	5	6	7	8	9
Amount of trees	2	37	45	11	4	1

node 3 is only 11 times divided into three subnodes. Interesting is also the fact that totally wrong endnodes are rarely determined. Only 4 of the trees perform "wrong" splits, like splitting node 5 further. Most of them simply merge two adjacent endnodes into one single endnode, for example they pool node 9 and node 10 into one single node.

5.3 Conclusion

The recursive partitioning algorithm allowing for non-binary splits has a strong power to detect the correct cutpoints and to grow well-balanced trees. In addition the non-binary splits are more concise than the binary equipollent. Especially the ability to grow nearly right-sized trees (it rather create less terminal nodes than too much) results in trees which presumably must not be pruned further.

Table 5: Endnodes detected by the trees with 6 endnodes

Nodenumber	5	6	7	8	9	10	4
Times this endnode occurs	44 (98%)	29 (64%)	31 (69%)	26 (58%)	11 (24%)	31 (69%)	45 (100%)

6 Application of the Methods

Finally we compare the enhanced version allowing for non-binary splits with the binary version by Breiman et al. [1] in a real data setting. We use a data set from a study of the DFG [6] about factors influencing the occurrence of chronic bronchitis. We assess the goodness of the non-binary tree using bootstrap methods. More informations about bootstraps can be found in Efron and Tibshirani [7].

6.1 With Binary Splits

First, a classification tree is grown using the dust data with CBR as outcome and time and overall dust as explanatory variables. We used standard settings in the growing and pruning process.

Using 10-fold cross-validation finally results in a tree with 6 endnodes with the following estimation of the misclassification rate:

Table 6: Resubstitution and cross-validation estimates of the misclassification rate $R^\star$ in the pruning process

Size	55	52	44	34	31
R^{RE}	0.182	0.183	0.186	0.191	0.194
R^{CV}	0.304	0.297	0.297	0.288	0.288
Deviance	715.0	721.3	742.8	775.9	785.3

Size	26	24	17	13	**6**	1
R^{RE}	0.198	0.200	0.211	0.217	**0.235**	0.262
R^{CV}	0.285	0.276	0.266	0.261	**0.258**	0.264
Deviance	818.8	829.4	865.0	900.0	**950.1**	1058.2

The tree is depicted in figure 6.

6.2 With non-binary Splits

The same data set is now analysed with the enhanced version of the CART algorithm. The result of this analysis is depicted in figure 7. Each of the boxes in 7 shows the number of people falling into this node and the contingent of people developing CBR. The number next to the boxes gives the node number. Again minimum node size was set to 5.

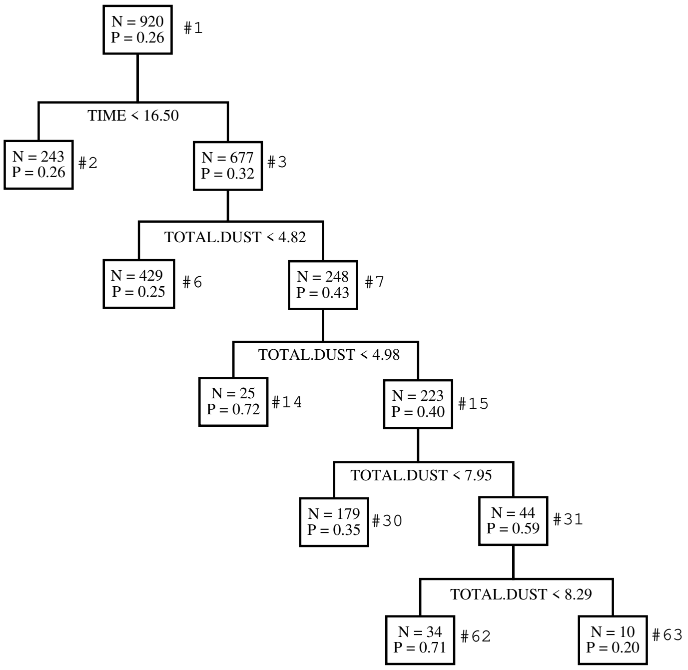

Figure 6: The final tree with six endnodes

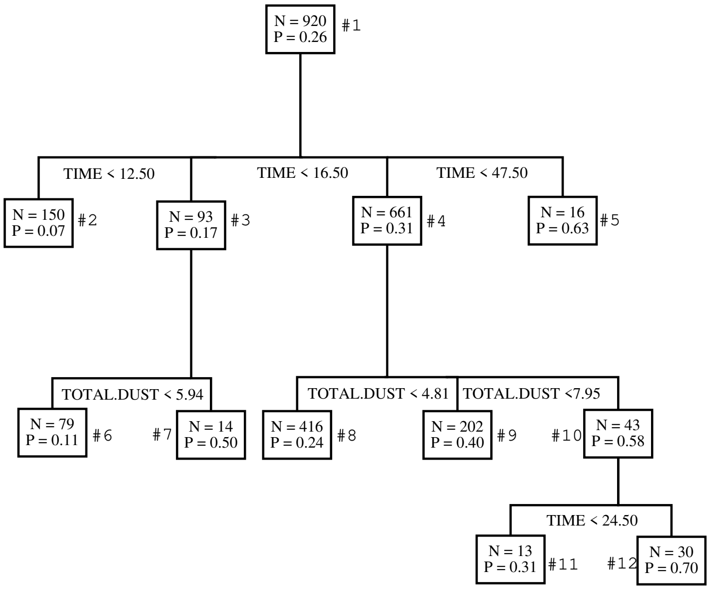

Figure 7: Tree allowing for non-binary splits

We can see in figure 7 that the enhanced version also finds the first split using the covariate TIME. One of the cutpoints is equal to the one of the CART algorithm at 16.5. Remarkable is also the analogy of the cutpoints leading to node 8 in the non-binary tree and to node 14 in the non-binary one, respectively to node 9

Table 7: Summary of cost-complexity pruning for the non-binary tree

Size	8	**7**	1
R^{RE}	0.2446	**0.2446**	0.2620
R^{boot}	0.2450	**0.2445**	0.2617
CI for R^{boot}	$[0.2163, 0.2750]$	$[\mathbf{0.2163},\ \mathbf{0.2739}]$	$[0.2348, 0.2913]$
Deviance	955.7522	**965.7054**	1058.172

and node 30. The same cutpoints (TIME < 16.5 and TOTAL.DUST < 4.8 resp. < 7.95) occur in *both* trees.

The resubstitution estimate for $R^\star$ for this tree is 0.245. Slightly higher than the one of the corresponding binary tree with 6 endnodes. A bootstrap estimator for $R^\star$ was found after drawing 1000 bootstrap samples as: $R^{boot} = 0.245$ with an 95%-confidence interval of $[0.216, 0.275]$.

We can also display our model as a class probability tree and give a three dimensional view of the model in figure 8.

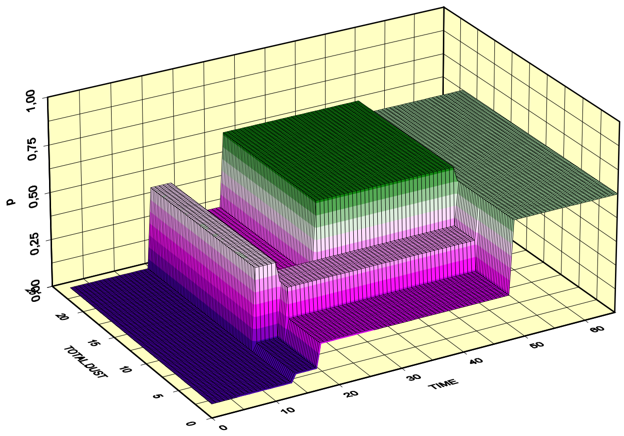

Figure 8: Three-dimensional plot of the class probability tree with non-binary splits corresponding to tree 7

6.3 Comparison

Both algorithms produce similar classifiers. As the CART algorithm is well established it seems like the two explanatory variables do not have strong predictive power on the occurrence of CBR. As both models finally do not perform well, the result do not differ much. Both find the split for TIME < 16.5 and then partition further using the amount of dust. The binary CART uses a sequence of dust splits to improve the performance which seems to be a bit artificial. In figure 9 we depict the branch at node 3 as a two dimensional plot. We can see that these partitions do not follow the monotonicity assumption. However, the relationship described in the branch does establish a dose-response relationship according to the trend tests Cochran-Armitage and the Isotonic Likelihood Ratio test which yield a p-value smaller than 0.001. For a more detailed description of these tests we refer to the paper by Salanti, Ulm [10].

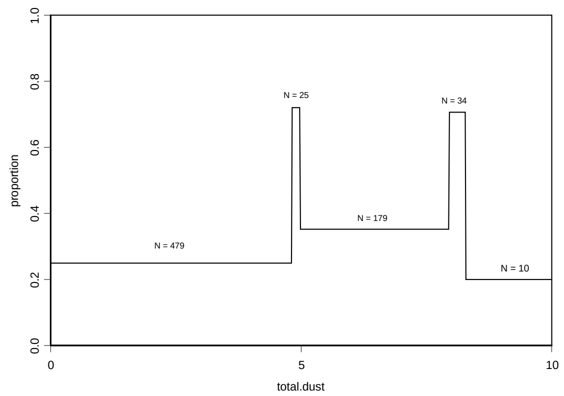

Figure 9: Two dimensional view upon the branch at node 3 of the binary tree with time > 16.5

On the other hand, the non-binary tree also finds splits for the group with TIME < 16.5 and leaves the impression of being better balanced in as sense of using cutpoints with a wider difference.

Finally, we give in a crosstable the number of classifications for both methods in table 8.

Table 8: Crosstable of the classifications

		Binary Tree	
		0	1
Multiple	0	816	44
Tree	1	35	25

7 Summary

In 1984 Breiman et al.[1] published a book called "Classification and Regression Trees", simply called "CART". In this book the well-known CART algorithm is introduced and several theoretical issues are discussed. The CART algorithm performs recursive partitioning and finally yields a decision tree. The most conspicuous character of this tree is the binary partitioning of a subgroup into two sibling nodes. In the following years many improvements to this algorithm have been proposed, but this binary splitting has been left mostly undiscussed. We propose a screening method which preselects the cutpoints of one covariate to a smaller amount. We applied this algorithm and substituted isotonic regression by its reduced version proposed by Salanti [9]. This version reduces the complexity of the models in isotonic regression, so that a more parsimonious model remains.

Isotonic regression is dependent on a monotonic assumption. In medical settings this assumption is associated with the so-called dose-response relationship. If this assumption does not hold this method should not be used. The isotonic regression results in a model which abstracts the observations into a number of different levels sets or solution blocks. These level sets can be represented by a step function. The exposure values corresponding to the beginning of new steps can be used as changepoints, which are then used as cutpoints in recursive partitioning. With PAVA, a fast and straightforward algorithm, we have a tool at hand, which yields the isotonic regression.

Isotonic regression can become more parsimonious if the levels sets associated with a non significant increase are merged together. Exact Fisher tests are used to decide which blocks should be amalgamated. As long as a p-value is higher than a certain value $\epsilon^\star$, the associated blocks are merged. Before starting the reduction $\epsilon^\star$ is calculated via simulations based on applying isotonic regression to permutations of the original dataset.

The main concept of recursive partitioning stays untouched if non-binary splits are permitted in CART. Still, some substantial modifications should be undertaken to ensure a satisfying tree growing. The computational effort grows quickly with the allowed number of possible daughter nodes. A simple comparison of all possible splits is soon getting too complex to be implemented in an algorithm. With the reduced isotonic regression we can find a small number of cutpoints in a manageable amount of time. The changepoints in reduced isotonic regression act as the cutpoints in the splitting of a node. Moreover, to get a fair comparisons of two or more different splits corresponding to different covariates, a generalised likelihood ratio test should be used. The splits belonging to the lowest p-value is declared as the overall best split of this node.

In the simulation study we paid special attention to the ability of the algorithm to detect the correct cutpoints and we showed that the algorithm does a good job in this field. It turned out that the reduced isotonic regression is, apart of being a good screening method, also a powerful and reasonable stopping rule.

We finally applied both recursive partitioning algorithms to a real dataset, the dataset from Munich. It arose that they yield similar trees.

It should be mentioned that the enhanced version is totally dependent on the monotonic assumption. If this assumption does not hold alternative changepoint detecting method should be tried or simply the conventional binary splitting be used. On the other hand, once a dose-response relationship can be established, the new recursive partitioning algorithm does use this knowledge and omits all splits violating this a-priori knowledge.

Concluding the reduced isotonic regression yields stable and reliable results and can be applied into a recursive partitioning algorithm. If a dose-response relationship is established, tree algorithm should and take advantage of it. It also turned out, that due to its unrestricted nature CART is more flexible to find a model that describes the data adequately than the enhanced version which is restricted by the isotonic assumption. In a situation with covariates with and without monotonic assumption a mixed tree algorithm can be tried, with likelihood ratio tests as goodness of split criteria.

References

[1] Breiman L., Friedman J.H., Olshen R.A. and Stone C.J. (1984) Classification and Regression Trees, *Chapman & Hall*, New York.

[2] Chambers J.M., Hastie T.J. (1993) Statistical Models in S, *Chapman & Hall*, New York.

[3] Barlow R.E., Bartholomew D.J., Bremner J.M., Brunk H.D. (1972) Statistical Inference under Order Restrictions, *Wiley*, New York.

[4] Dannegger F. (1994) Eine Erweiterung der CART-Methode auf Mehrweg-Splits, *Diplomarbeit*, University Munich.

[5] Dannegger F., (1997) Improving Predictions of Tree-Structured Algorithms, *Dissertation*, University Munich.

[6] Deutsche Forschungsgemeinschaft: DFG Research Report (1978) Chronic Bronchitis, *Harald Boldt*, Boppard.

[7] Efron B. and Tibshirani R.J. (1993) An Introduction to the Bootstrap, *Chapman & Hall*, New York.

[8] Robertson T., Wright F. and Dykstra R.L. (1988) Order Restricted Statistical Inference, *John Wiley & Sons*, New York.

[9] Salanti G. (2003) The isotonic regression framework - Estimating and testing under order restrictions, *Dissertation*, University Munich.

[10] Salanti G. and Ulm K. (2003) Tests for Trend with Binary Response, *Biometrical Journal 43, 277-291.*

[11] Schell M.J. and Singh B. (1997) The Reduced Monotonic Regression Method, *American Statistical Association, 92:128:135.*

[12] Strobl R. (2003) Multiple Splitting under Order Restrictions in Classification Trees, *Diplomarbeit*, University Munich.

[13] Zhang H. and Singer B. (1999) Recursive Partitioning in the Health Science, *Springer*, New York.