Classification Trees

Data Mining 2015: assignment 1

Sebastiaan Jong (5546303) & Bas Geerts (5568978)

1 Introduction

This report is written for the first assignment of the Data Mining (2015) course at Utrecht University. The goal of this assignment was to write a function in the R programming language that constructs a classification tree on a certain dataset, and to figure out efficient parameters for this tree.

2 Data

For this assignment we used the Heart Disease dataset from the University of California, Irvine machine learning repository. The preprocessed version for this assignment contains 297 instances and 14 attributes. The class label for this assignment is AHD, which indicates whether the patient has been diagnosed with a heart disease, the preprocessed dataset contains 137 instances where this is the case.

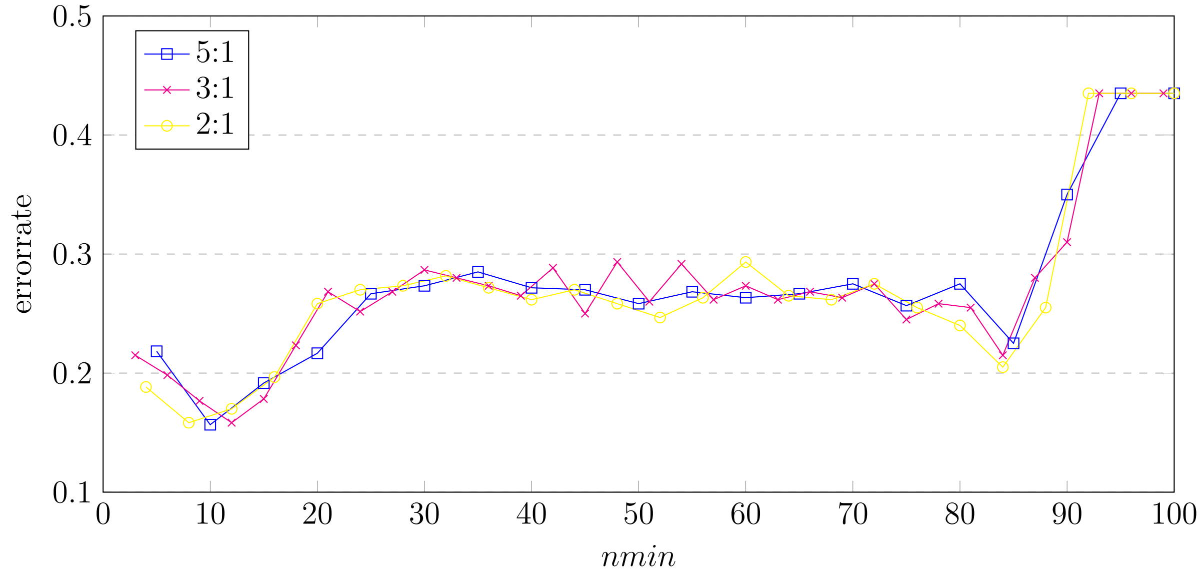
3 Experiments

The size of the classification tree is controlled by the *nmin* (minimum internal node size) and *minleaf* (minimum leaf size) parameters. Instead of brute forcing all possible values it is better to only try some sensible values and narrow down the optimal settings from there. It is possible to make a few observations regarding sensible parameter values (where n is the dataset size):

- Both $nmin$ and $minleaf$ should not be larger than $n/2$.
- The $minleaf$ parameter should not be larger than $nmin/2$.

The experiment uses a random trainingset of 200 rows from the initial dataset, the remaining 97 rows are used as testsample. To quickly find an approximation of good parameter values, we tried all likely values for *nmin* between 1 and 100 with a certain ratio to $minleaf$. The results are plotted in Figure 1, results are obtained by performing a 10-fold cross validation for each different parameter on the same trainingset. For example, the ratio 3:1 means that at $nmin = 45$, we tried $minleaf = 15$. The results from Figure 1 indicate that a well performing value for $nmin$ might be found between 10 and 20, since the error rate is consistently low at these points.

Figure 1. *The nmin:minleaf ratios tried*



The results of performing another cross validation on all $nmin$ and $minleaf$ values where $nmin$ is between 8 and 20 with steps of 2 is shown in table 1. The values for $nmin$ 8, 18 and 20 have been omitted since they all scored well above an errorrate of 0.20. The columns errorrate, nodes and leaves indicate the average of the 10 trees created during the cross validation.

In table 1 the parameter pairs $(16, 8), (16, 4)$ and $(12, 2)$ have the lowest errorrate, with $(16, 8)$ also having a relatively small tree. Constructing a tree with these parameters on the trainingset, and predicting the classes of the testsample gives us the results as shown in table 2. The data in table 2 shows us that $nmin = 16$ and $minlead = 8$ are the best parameters, since they have the lowest errorrate and provide the smallest tree.

nmin	*minleaf*	errorrate	nodes	leaves	runtime(sec)
16	8	0.16	6.4	7.4	1.41
16	6	0.20	7.5	8.5	1.47
16	4	0.16	9.4	10.4	1.60
16	2	0.18	12.1	13.1	1.74
14	7	0.21	7.8	8.8	1.50
14	5	0.15	8.9	9.9	1.57
14	3	0.19	11.6	12.6	1.79
14	1	0.24	17.2	18.2	1.96
12	6	0.17	9.4	10.4	1.60
12	4	0.19	10.6	11.6	1.65
12	2	0.15	12.8	13.8	1.74
10	5	0.17	10.8	11.8	1.66
10	3	0.18	12.4	13.4	1.72
10	1	0.22	18.4	19.4	1.96

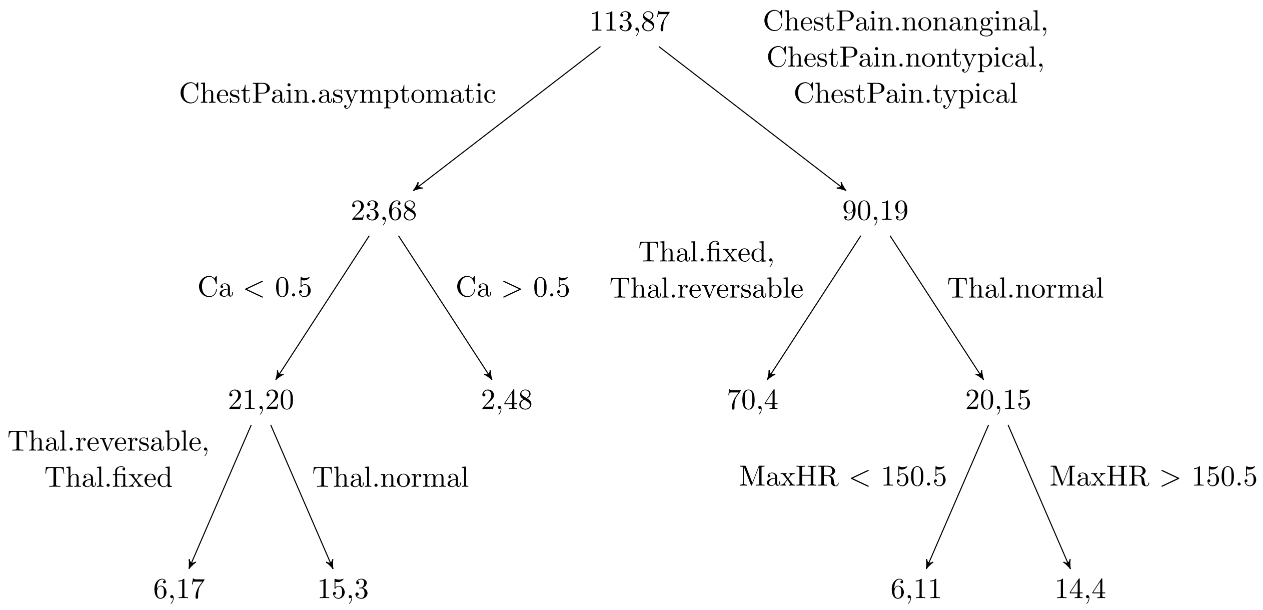
Table 1: *Crossvalidation results.*

nmin	*minleaf*	errorrate	nodes	leaves
16	8	0.21	8	9
16	4	0.24	9	10
12	2	0.26	15	16

Table 2: *Trees grown on the complete trainingset.*

4 Results

The final tree, constructed with the parameters $nmin = 16$ and $minleaf = 8$ can be seen in figure 2. Splits are shown in the labels of edges. The numbers in the nodes mean the amount of instances labeled '0' and the amount of instances labeled '1' respectively. An obvious problem with this tree is the leaf with 6 zero-labeled instances and 11 one-labeled instances. The resubstitution error of this node is $1 - (11/17) \approx 0.35$, which is fairly high. A confusion matrix can be found in table 3, the estimated errorrate is 0.21.



	0	1
0	42	16
1	5	34

Table 3: *Confusion matrix of the final tree.*