Unbiased Recursive Partitioning I: A Non-parametric Conditional Inference Framework

Torsten Hothorn[1], **Kurt Hornik**[2] **and Achim Zeileis**[2]

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg

[2] Wirtschaftsuniversität Wien

Recursive Partitioning

Morgan and Sonquist (1963): Automated Interaction Detection.

Many variants have been (and still are) published, the majority of which are special cases of a simple two-stage algorithm:

Step 1: partition the observations by univariate splits in a recursive way

Step 2: fit a constant model in each cell of the resulting partition.

Most prominent representatives: 'CART' (Breiman et al., 1984) and 'C4.5' (Quinlan, 1993), both implementing an exhaustive search.

Two Fundamental Problems

Overfitting: Mingers (1987) notes that the algorithm

> [...] has no concept of statistical significance, and so cannot distinguish between a significant and an insignificant improvement in the information measure.

Selection Bias: Exhaustive search methods suffer from a severe selection bias towards covariates with many possible splits and / or missing values.

A Statistical Approach

We enter at the point where White and Liu (1994) demand for

> [...] a *statistical* approach [to recursive partitioning] which takes into account the *distributional* properties of the measures.

and present a unified framework embedding recursive binary partitioning into the well defined theory of

Part I: permutation tests developed by Strasser and Weber (1999),

Part II: tests for parameter instability in (parametric) regression models.

The Regression Problem

The distribution of a (possibly multivariate) response $\mathbf{Y} \in \mathcal{Y}$ is to be regressed on a m-dimensional covariate vector $\mathbf{X} = (X_1, \ldots, X_m) \in \mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_m$:

$$D(\mathbf{Y}|\mathbf{X}) = D(\mathbf{Y}|X_1, \ldots, X_m) = D(\mathbf{Y}|f(X_1, \ldots, X_m)),$$

based on a learning sample of n observations

$$\mathcal{L}_n \quad = \quad \{(\mathbf{Y}_i, X_{1i}, \ldots, X_{mi}); i = 1, \ldots, n\}.$$

possibly with case counts $\mathbf{w} = (w_1, \ldots, w_n)$. We are interested in estimating f.

A Generic Algorithm

1. For case weights $\mathbf{w}$ test the global null hypothesis of independence between any of the m covariates and the response. Stop if this hypothesis cannot be rejected. Otherwise select the covariate X_{j^*} with strongest association to $\mathbf{Y}$.

2. Choose a set $A^* \subset \mathcal{X}_{j^*}$ in order to split $\mathcal{X}_{j^*}$ into two disjoint sets A^* and $\mathcal{X}_{j^*} \setminus A^*$. The case weights $\mathbf{w}_{\text{left}}$ and $\mathbf{w}_{\text{right}}$ determine the two subgroups with $w_{\text{left},i} = w_i I(X_{j^*i} \in A^*)$ and $w_{\text{right},i} = w_i I(X_{j^*i} \notin A^*)$ for all $i = 1, \ldots, n$ ($I(\cdot)$ denotes the indicator function).

3. Recursively repeat steps 1 and 2 with modified case weights $\mathbf{w}_{\text{left}}$ and $\mathbf{w}_{\text{right}}$, respectively.

Recursive Partitioning by Conditional Inference

In each node identified by case weights $\mathbf{w}$, the global hypothesis of independence is formulated in terms of the m partial hypotheses $H_0^j : D(\mathbf{Y}|X_j) = D(\mathbf{Y})$ with global null hypothesis $H_0 = \bigcap_{j=1}^{m} H_0^j$.

Stop recursion when H_0 can *not* be rejected at a pre-specified level α.

Else: Measure the association between $\mathbf{Y}$ and each of the covariates $X_j, j = 1, \ldots, m$, by test statistics or P-values indicating the deviation from the partial hypotheses H_0^j.

Linear Statistics

Use a (multivariate) linear statistic

$$\mathbf{T}_j(\mathcal{L}_n, \mathbf{w}) = \text{vec}\left(\sum_{i=1}^{n} w_i g_j(X_{ji}) h(\mathbf{Y}_i, (\mathbf{Y}_1, \ldots, \mathbf{Y}_n))^\top\right) \in \mathbb{R}^{p_j q}$$

with g_j being a transformation of X_j and *influence function* $h : \mathcal{Y} \times \mathcal{Y}^n \to \mathbb{R}^q$. This type of statistics was suggested by Strasser and Weber (1999).

Conditional Expectation and Covariance under H_0^j

$$
\begin{aligned}
\mu_j &= \mathbb{E}(\mathbf{T}_j(\mathcal{L}_n, \mathbf{w})|S(\mathcal{L}_n, \mathbf{w})) = \mathsf{vec}\left(\left(\sum_{i=1}^n w_i g_j(X_{ji})\right)\mathbb{E}(h|S(\mathcal{L}_n, \mathbf{w}))^\top\right), \\
\Sigma_j &= \mathbb{V}(\mathbf{T}_j(\mathcal{L}_n, \mathbf{w})|S(\mathcal{L}_n, \mathbf{w})) \\
&= \frac{\mathbf{w}.}{\mathbf{w}. - 1}\mathbb{V}(h|S(\mathcal{L}_n, \mathbf{w})) \otimes \left(\sum_i w_i g_j(X_{ji}) \otimes w_i g_j(X_{ji})^\top\right) \\
&- \frac{1}{\mathbf{w}. - 1}\mathbb{V}(h|S(\mathcal{L}_n, \mathbf{w})) \otimes \left(\sum_i w_i g_j(X_{ji})\right) \otimes \left(\sum_i w_i g_j(X_{ji})\right)^\top
\end{aligned}
$$

with $\mathbf{w}. = \sum_{i=1}^n w_i$.

Test Statistics

A (multivariate) linear statistic $\mathbf{T}_j$ can now be used to construct a test statistic for testing H_0^j, for example via

$$c_{\text{max}}(\mathbf{t}, \mu, \Sigma) = \max_{k=1,\ldots,pq} \left| \frac{(\mathbf{t} - \mu)_k}{\sqrt{(\Sigma)_{kk}}} \right|$$

or

$$c_{\text{quad}}(\mathbf{t}, \mu, \Sigma) = (\mathbf{t} - \mu)\Sigma^+(\mathbf{t} - \mu)^\top$$

Variable Selection and Stopping Criteria

Test H_0 based on $P_1, \ldots, P_m$,

$$P_j = \mathbb{P}_{H_0^j}(c(\mathbf{T}_j(\mathcal{L}_n, \mathbf{w}), \mu_j, \Sigma_j) \geq c(\mathbf{t}_j, \mu_j, \Sigma_j) | S(\mathcal{L}_n, \mathbf{w}))$$

conditional on all permutations $S(\mathcal{L}_n, \mathbf{w})$ of the data. *This solves the overfitting problem*.

When we can reject H_0 in step 1 of the generic algorithm we select the covariate with minimum P-value

$$P_j = \mathbb{P}_{H_0^j}(c(\mathbf{T}_j(\mathcal{L}_n, \mathbf{w}), \mu_j, \Sigma_j) \geq c(\mathbf{t}_j, \mu_j, \Sigma_j) | S(\mathcal{L}_n, \mathbf{w}))$$

of the conditional test for H_0^j. *This prevents a variable selection bias*.

Splitting Criteria

The goodness of a split is evaluated by two-sample linear statistics which are special cases of the linear statistic $\mathbf{T}$. For all possible subsets A of the sample space $\mathcal{X}_{j^*}$ the linear statistic

$$\mathbf{T}_{j^*}^A(\mathcal{L}_n, \mathbf{w}) = \mathsf{vec}\left(\sum_{i=1}^n w_i I(X_{j^*i} \in A) h(\mathbf{Y}_i, (\mathbf{Y}_1, \ldots, \mathbf{Y}_n))^\top\right)$$

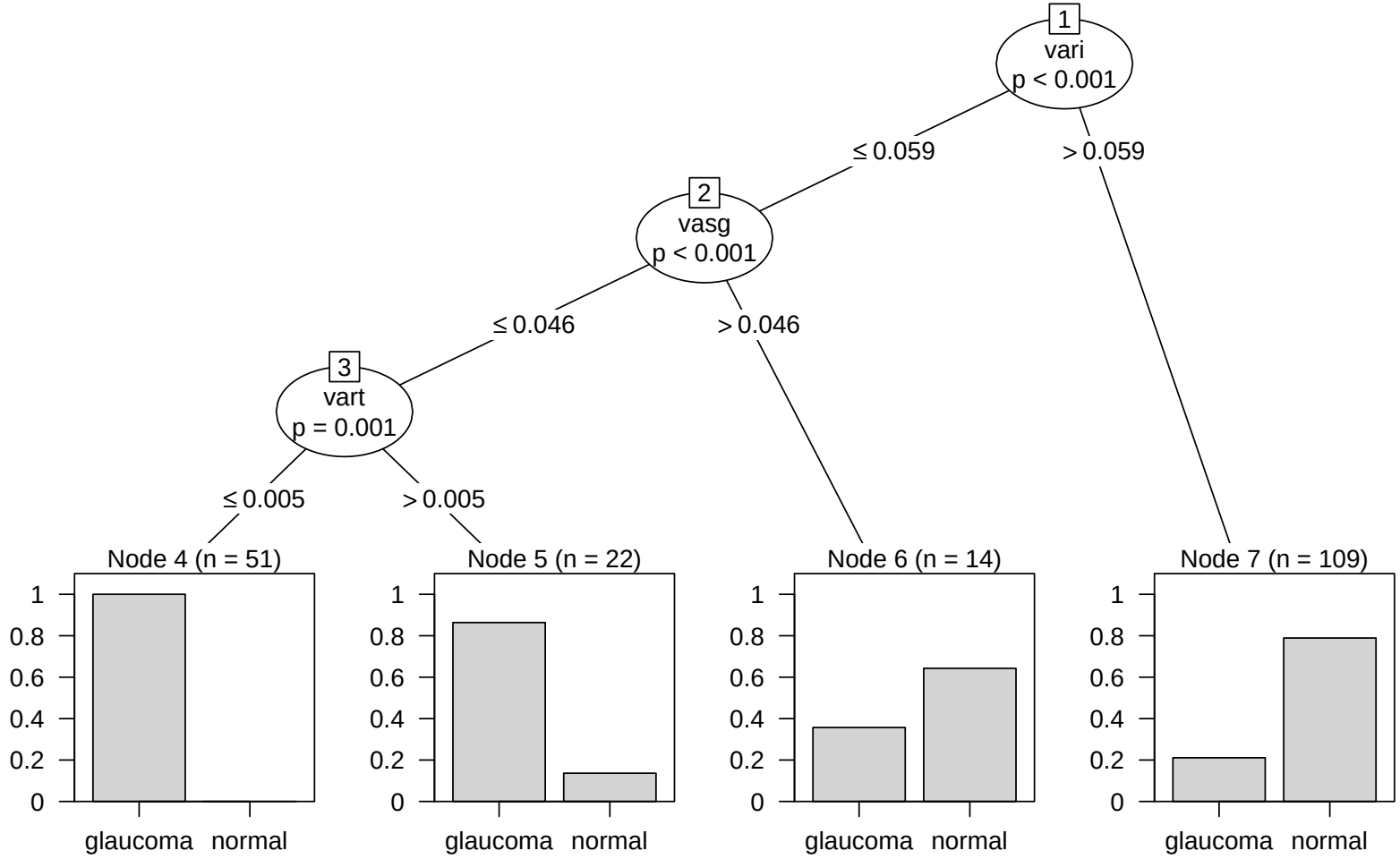
induces a two-sample statistic and we implement the best split

$$A^* = \mathsf{argmax}_A c(\mathbf{t}_{j^*}^A, \mu_{j^*}^A, \Sigma_{j^*}^A).$$

Examples: Glaucoma & Laser Scanning Images

Laser scanning images taken from the eye background are expected to serve as the basis of an automated system for glaucoma diagnosis. For 98 patients and 98 controls, matched by age and gender, 62 covariates describing the eye morphology are available.

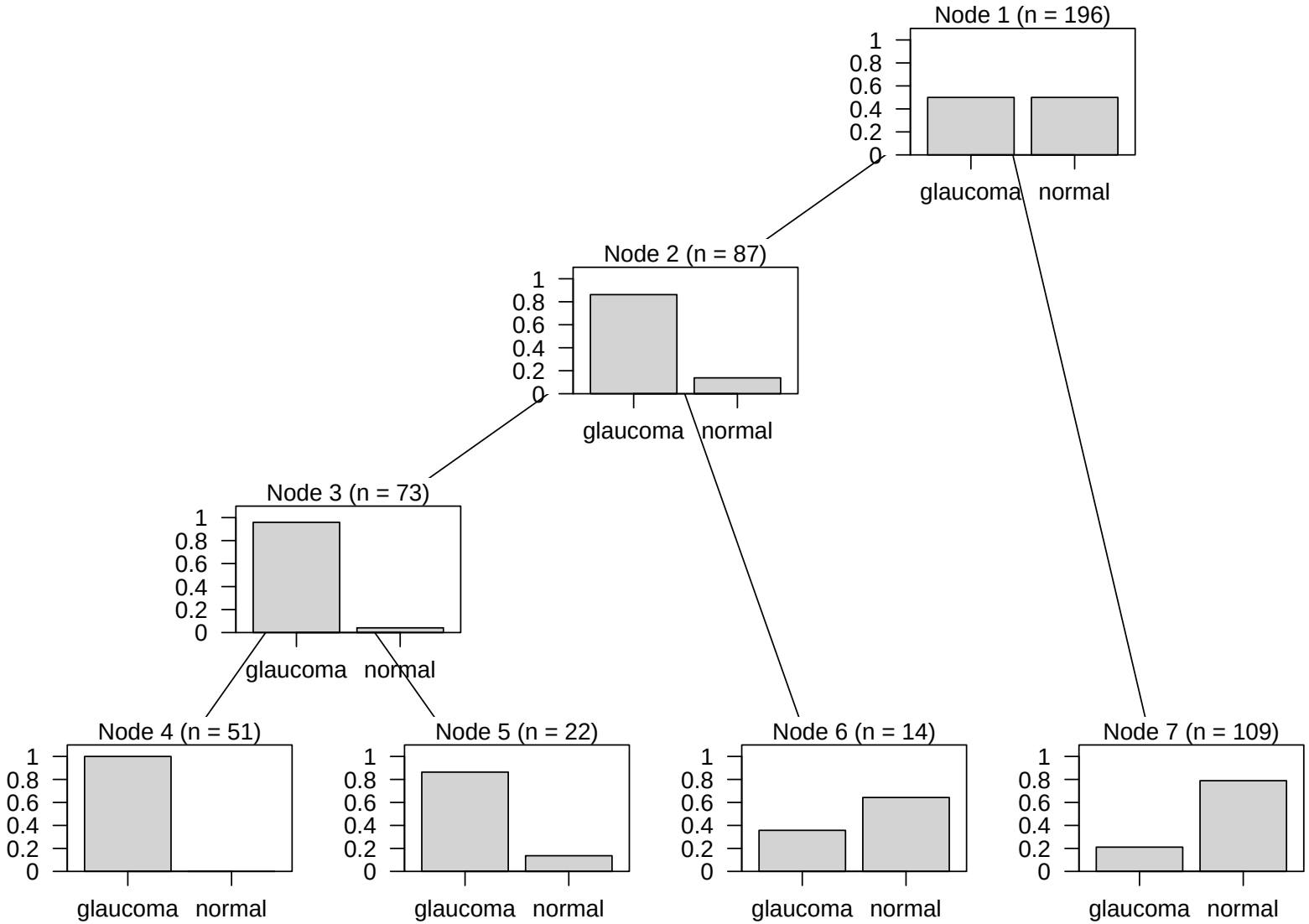
```
R> data(GlaucomaM, package = "ipred")
R> gtree <- ctree(Class ~ ., data = GlaucomaM)
R> plot(gtree)
```


1
vari
p < 0.001
≤ 0.059
> 0.059
2
vasg
p < 0.001
≤ 0.046
> 0.046
3
vart
p = 0.001
≤ 0.005
> 0.005
Node 4 (n = 51)
Node 5 (n = 22)
Node 6 (n = 14)
Node 7 (n = 109)
1
0.8
0.6
0.4
0.2
0
glaucoma
normal

Examples: Glaucoma & Laser Scanning Images

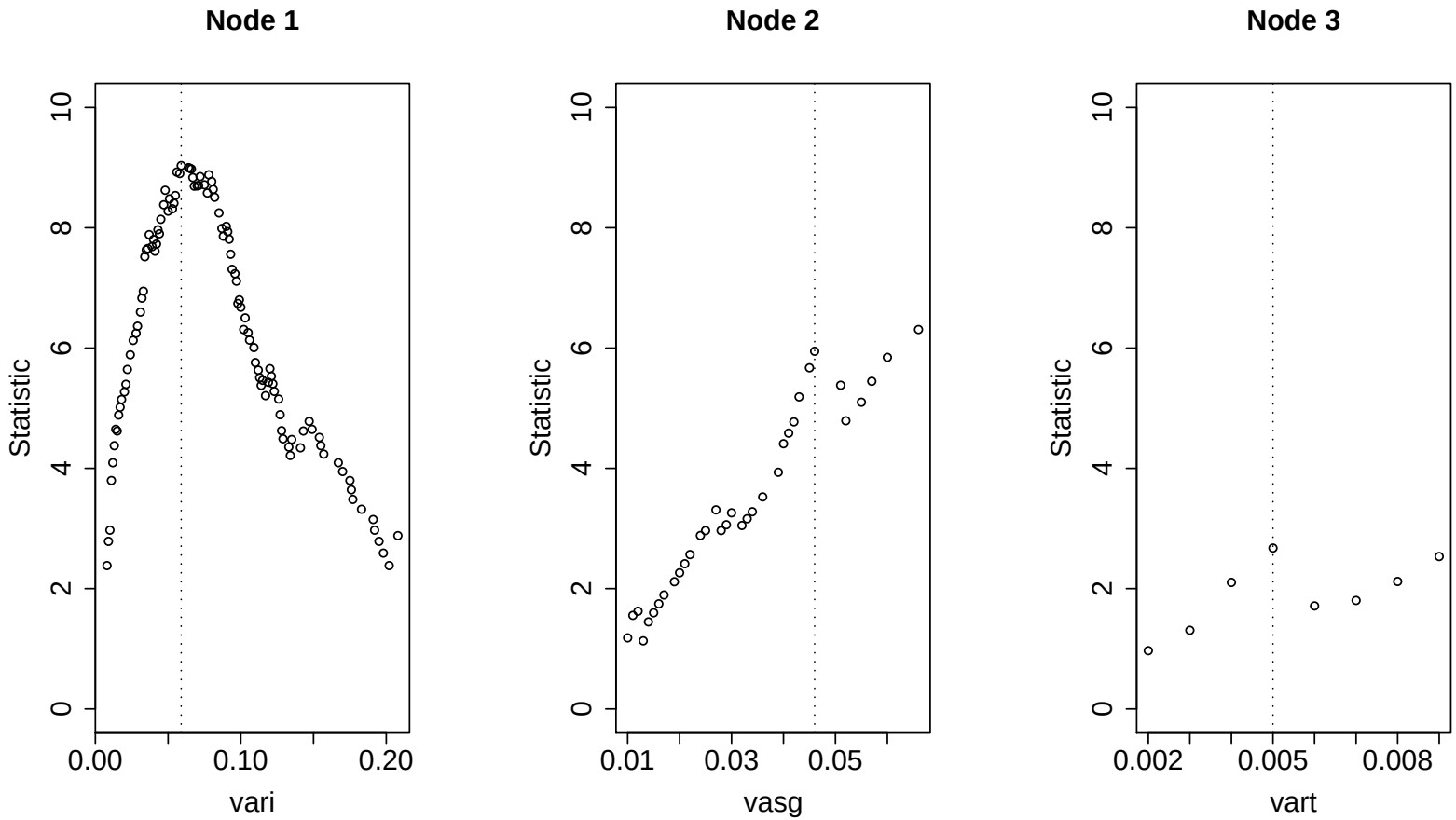
Interested in the class distribution in each *inner* node? Want to explore the process of the split statistics in each inner node?

```
R> plot(gtree, inner_panel = node_barplot, tnex = 1)
```


Node 1 (n = 196)
1
0.8
0.6
0.4
0.2
0
glaucoma
normal
Node 2 (n = 87)
1
0.8
0.6
0.4
0.2
0
glaucoma
normal
Node 3 (n = 73)
1
0.8
0.6
0.4
0.2
0
glaucoma
normal
Node 4 (n = 51)
1
0.8
0.6
0.4
0.2
0
glaucoma
normal
Node 5 (n = 22)
1
0.8
0.6
0.4
0.2
0
glaucoma
normal
Node 6 (n = 14)
1
0.8
0.6
0.4
0.2
0
glaucoma
normal
Node 7 (n = 109)
1
0.8
0.6
0.4
0.2
0
glaucoma
normal

Examples: Glaucoma & Laser Scanning Images

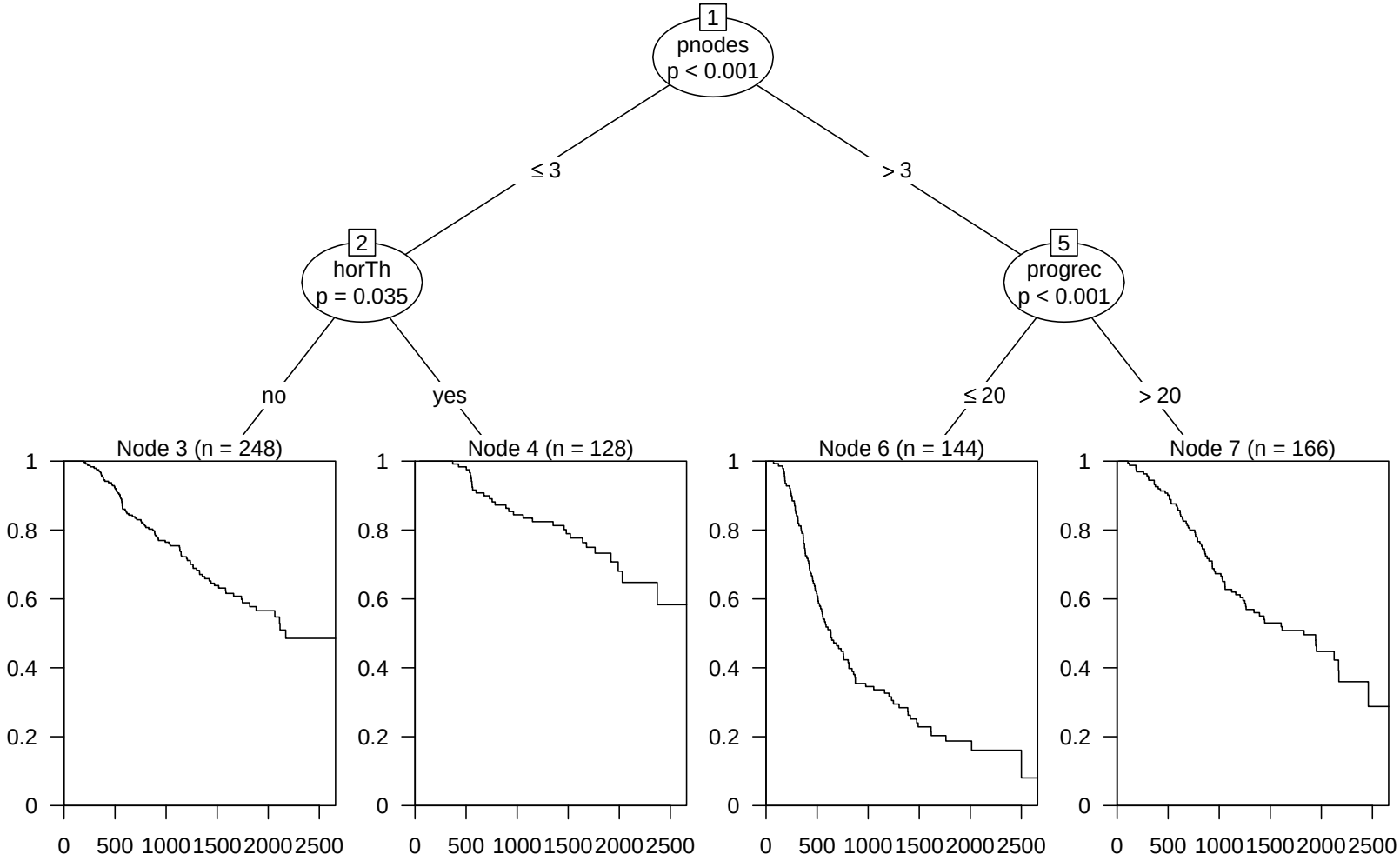
```
R> cex <- 1.6
R> inner <- nodes(gtree, 1:3)
R> layout(matrix(1:length(inner), ncol = length(inner)))
R> out <- sapply(inner, function(i) {
+     splitstat <- i$psplit$splitstatistic
+     x <- GlaucomaM[[i$psplit$variableName]][splitstat >
+         0]
+     plot(x, splitstat[splitstat > 0], main = paste("Node",
+         i$nodeID), xlab = i$psplit$variableName,
+         ylab = "Statistic", ylim = c(0, 10),
+         cex.axis = cex, cex.lab = cex, cex.main = cex)
+     abline(v = i$psplit$splitpoint, lty = 3)
+ })
```


Node 1
Statistic
0
2
4
6
8
10
0.00
0.10
0.20
vari
Node 2
Statistic
0
2
4
6
8
10
0.01
0.03
0.05
vasg
Node 3
Statistic
0
2
4
6
8
10
0.002
0.005
0.008
vart

Examples: Node Positive Breast Cancer

Evaluation of prognostic factors for the German Breast Cancer Study Group (GBSG2) data, a prospective controlled clinical trial on the treatment of node positive breast cancer patients. Complete data of seven prognostic factors of 686 women are used for prognostic modeling

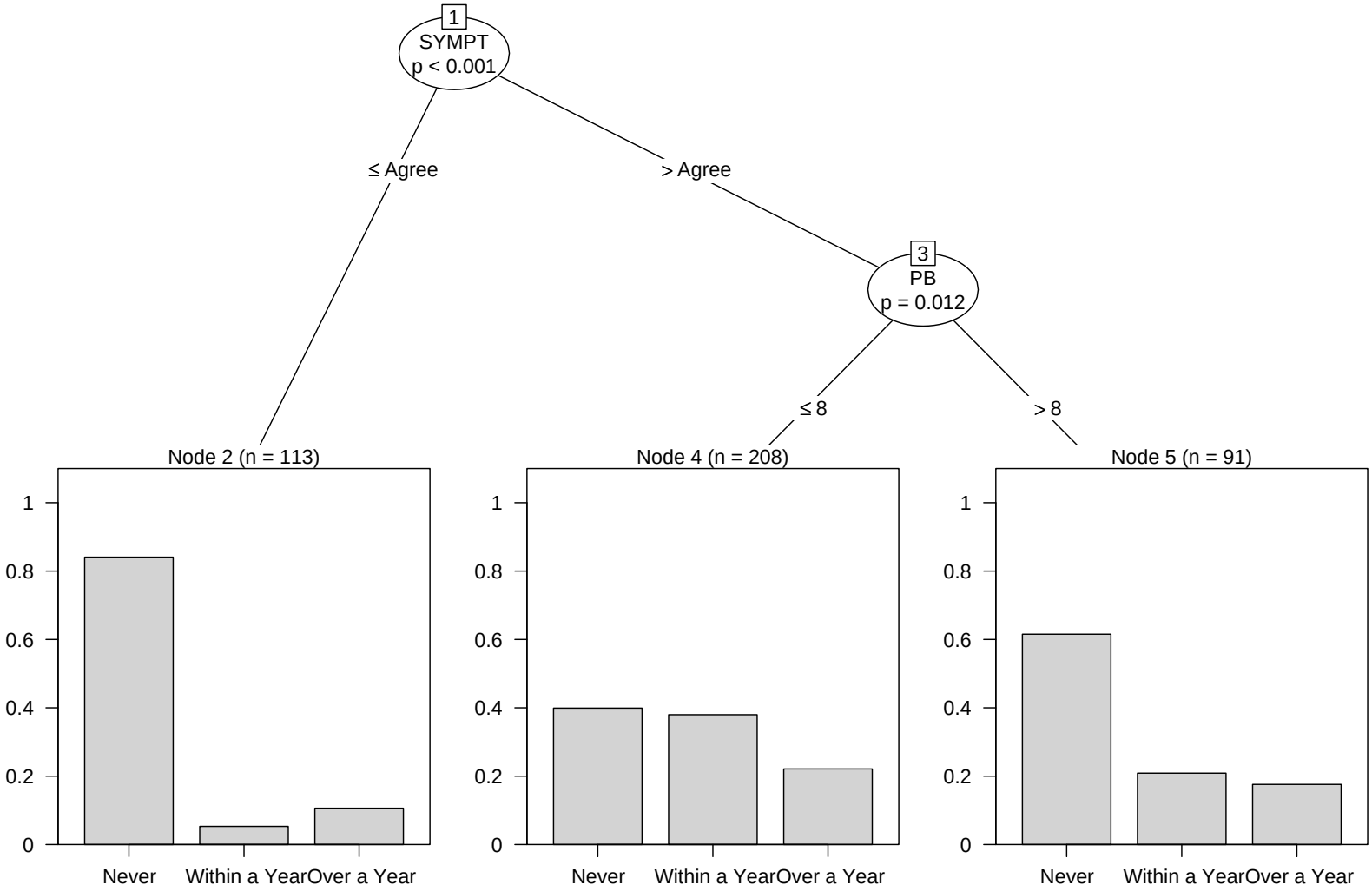
```
R> data(GBSG2, package = "ipred")
R> stree <- ctree(Surv(time, cens) ~ ., data = GBSG2)
R> plot(stree)
```


1
pnodes
p < 0.001
≤ 3
> 3
2
horTh
p = 0.035
5
progrec
p < 0.001
no
yes
≤ 20
> 20
Node 3 (n = 248)
Node 4 (n = 128)
Node 6 (n = 144)
Node 7 (n = 166)
1
0.8
0.6
0.4
0.2
0
0 500 1000 1500 2000 2500

Examples: Mammography Experience

Ordinal response variables are common in investigations where the response is a subjective human interpretation. We use an example given by Hosmer and Lemeshow (2000), p. 264, studying the relationship between the mammography experience (never, within a year, over one year) and opinions about mammography expressed in questionnaires answered by $n = 412$ women.

```
R> data(mammoexp, package = "party")
R> mtree <- ctree(ME ~ ., data = mammoexp, scores = list(ME = 1:3,
+     SYMPT = 1:4, DECT = 1:3))
R> plot(mtree)
```


1
SYMPT
p < 0.001
≤ Agree
> Agree
3
PB
p = 0.012
≤ 8
> 8
Node 2 (n = 113)
Node 4 (n = 208)
Node 5 (n = 91)
1
0.8
0.6
0.4
0.2
0
Never
Within a Year
Over a Year

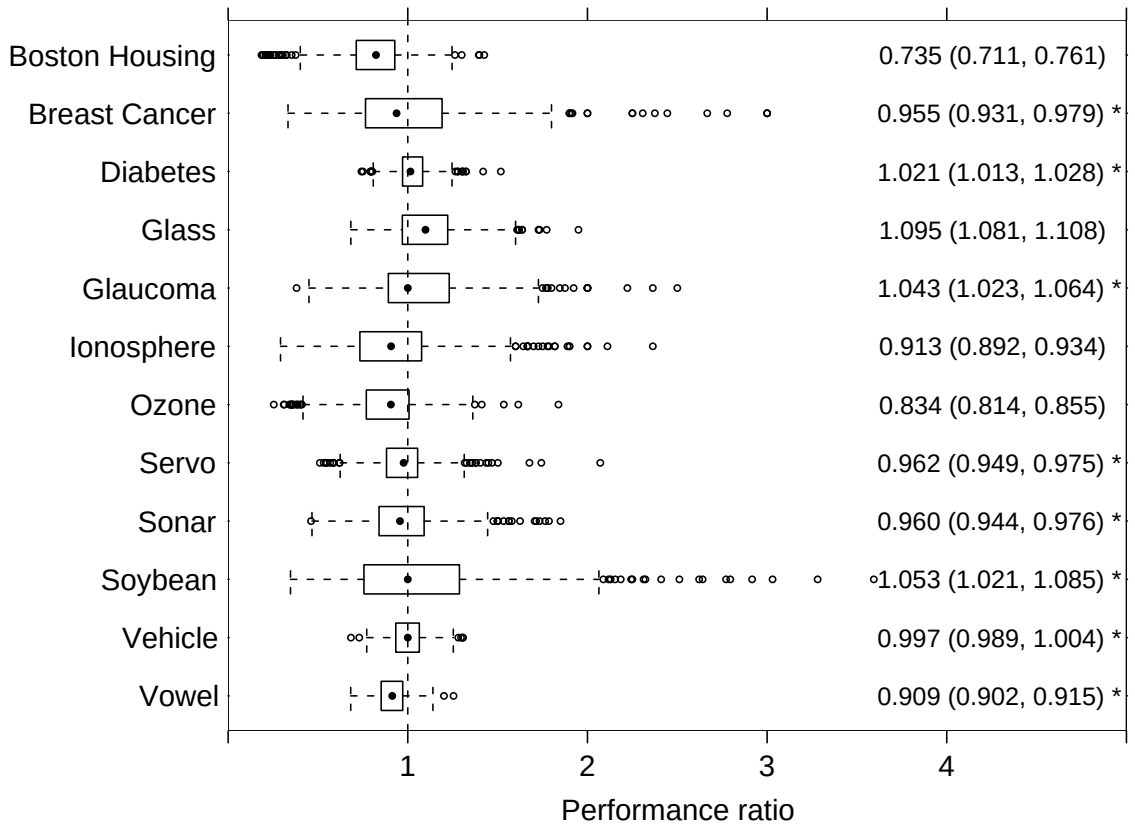
Benchmark Experiments

Hypothesis 1: Conditional inference trees with statistical stop criterion perform as good as an exhaustive search algorithm with pruning.

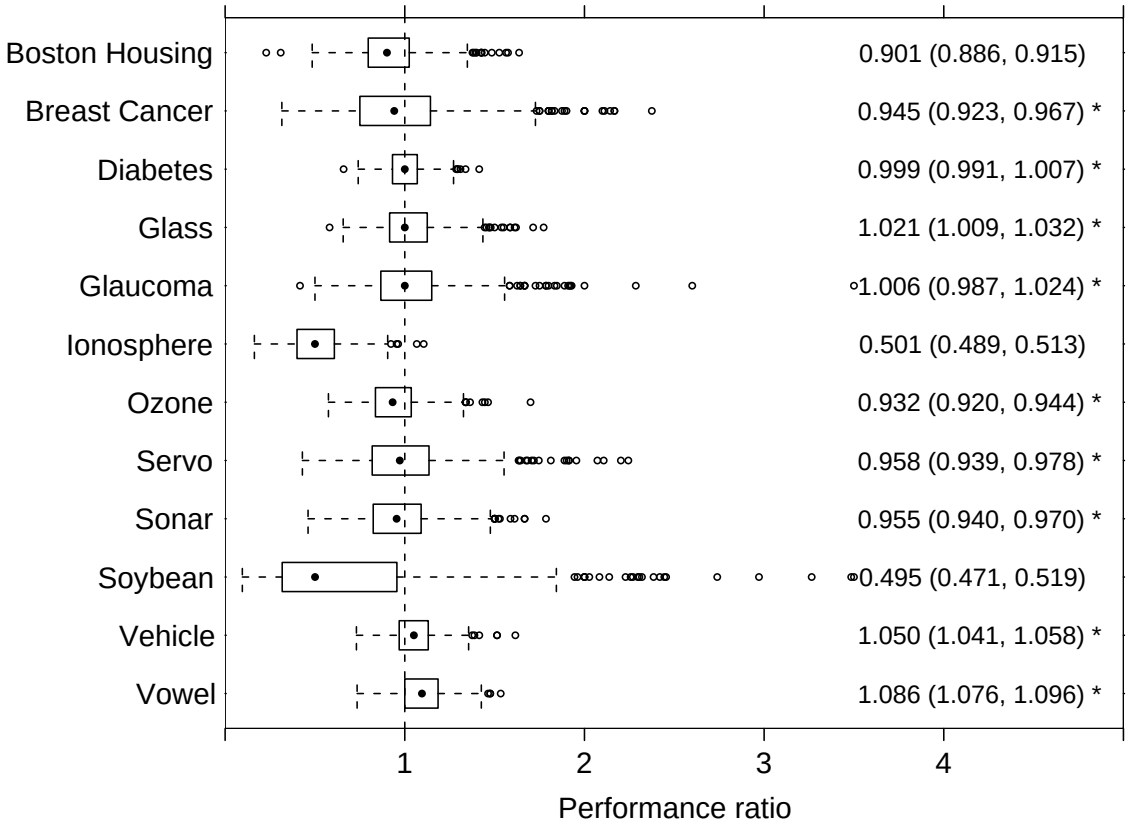
Hypothesis 2: Conditional inference trees with statistical stop criterion perform as good as parametric unbiased recursive partitioning (QUEST,GUIDE, Loh, 2002, is a starting point).

Equivalence measured by ratio of misclassification or mean squared error with a equivalence margin of $\pm 10\%$ and Fieller confidence intervals.

ctree **vs.** rpart



ctree **vs.** QUEST / GUIDE



Summary

The separation of variable selection and split point estimation first implemented in 'CHAID' (Kass, 1980) is the basis for unbiased recursive partitioning for responses and covariates measured at arbitrary scales.

The statistical internal stop criterion ensures that interpretations drawn from such trees are valid in a statistical sense, i.e., with appropriate control of type I errors.

Even the algorithm has no concept of prediction error, the performance is at least equivalent to established procedures.

We are committed to reproducible research, see

```
R> vignette(package = "party")
```

References

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. Wadsworth, California, 1984.

David W. Hosmer and Stanley Lemeshow. *Applied Logistic Regression*. John Wiley & Sons, New York, 2nd edition, 2000.

G.V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29(2):119–127, 1980.

Wei-Yin Loh. Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, 12:361–386, 2002.

John Mingers. Expert systems – rule induction with statistical data. *Journal of the Operations Research Society*, 38(1):39–47, 1987.

James N. Morgan and John A. Sonquist. Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58:415–434, 1963.

John R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publ., San Mateo, California, 1993.

Helmut Strasser and Christian Weber. On the asymptotic theory of permutation statistics. *Mathematical Methods of Statistics*, 8:220–250, 1999.

Allan P. White and Wei Zhong Liu. Bias in information-based measures in decision tree induction. *Machine Learning*, 15:321–329, 1994.