

SUPPLEMENT TO

The potential of model-based recursive partitioning in the social sciences – Revisiting Ockham’s Razor

Julia Kopf, Thomas Augustin, Carolin Strobl

This supplement¹ is designed to illustrate applications of classification and regression trees as well as model-based recursive partitioning in the freely available software **R** (R Development Core Team 2009; see, e.g., Dalgaard 2002, for a general introduction to R).

The examples presented in this supplement are similar in spirit to the examples presented in the chapter, but simulated data are used. For terms to access the German Socio-Economic Panel Study (SOEP) from 2008, provided by DIW Berlin (German Institute for Economic Research), visit <http://www.diw.de/>.

Preliminary settings

- Upload the file `example_data.RData` (an active internet connection is necessary)

```
> load(url("http://www.statistik.lmu.de/~kopf/example_data.RData"))
```

The file contains three data sets, namely `dat_empl`, `dat_unempl`, `dat_job`.

- Inspect data structure, e.g. for `dat_empl` containing `gender`, `age`, `job_time`

```
> names(dat_empl)
```

```
[1] "gender"    "age"       "job_time"
```

```
> head(dat_empl)
```

```
  gender age job_time
1  male  31    other
2  male  53 full time
3  male  36    other
4  male  57 full time
5  male  59    other
6  male  20    other
```

```
> str(dat_empl)
```

```
'data.frame':      19553 obs. of  3 variables:
 $ gender  : Factor w/ 2 levels "male","female": 1 1 1 1 1 1 1 1 1 1 ...
 $ age     : num  31 53 36 57 59 20 41 57 42 38 ...
 $ job_time: Factor w/ 2 levels "full time","other": 2 1 2 1 2 2 1 1 1 2 ...
```

```
> summary(dat_empl)
```

¹This document was created with **Sweave** to combine **L^AT_EX** and R code (Leisch 2002).

2 SUPPLEMENT

gender	age	job_time
male : 9318	Min. :18.00	full time: 7404
female:10235	1st Qu.:33.00	other :12149
	Median :48.00	
	Mean :45.98	
	3rd Qu.:62.00	
	Max. :64.00	

- Load package `party` (Hothorn, Hornik, & Zeileis 2006; Zeileis, Hothorn, & Hornik 2008) (if `party` has not yet been used, the `install.packages()` command allows to download the package)

```
> install.packages("party", dependencies=TRUE)
> library("party")
```

Classification tree

- Construct a classification tree for a binary response variable (here: `job_time`)

```
> ct_obj <- ctree(job_time ~ gender + age,
>                 control = ctree_control(minsplit = 50), data = dat_empl)
```
- Examine the resulting classification tree object

```
> ct_obj
```

Conditional inference tree with 4 terminal nodes

Response: `job_time`

Inputs: `gender`, `age`

Number of observations: 19553

```
1) gender == {male}; criterion = 1, statistic = 1910.231
  2) age <= 62; criterion = 1, statistic = 1397.736
    3)* weights = 6835
  2) age > 62
    4)* weights = 2483
1) gender == {female}
  5) age <= 60; criterion = 1, statistic = 530.524
    6)* weights = 7274
  5) age > 60
    7)* weights = 2961
```

Here, `job_time` is a binary response variable – indicating whether or not the respondent works full-time ("`full time`", "`other`") – whereas `gender` and `age` are potential splitting variables like in the example in section 2.1. Splitting variables can be of every commonly used covariate format (factor, ordered factor or numerical) as long as it is correctly specified in the software. The argument `control = ctree_control()` offers various options the user can specify: `minsplit=50` requires at least 50 observations in a node to conduct a split. Alpha level, test statistics and distributive characteristics can be chosen as well (see `?ctree` in R).

- Plot the `ctree`-object

```
> plot(ct_obj, terminal_panel = node_barplot(ct_obj, beside=TRUE))
```

Calling `plot(ct_obj)` would result in a stacked bar plot, while the option `terminal_panel = node_barplot(ct_obj, beside=TRUE)` displays a vertical-bar chart such as in Figure 3.1 in the chapter.

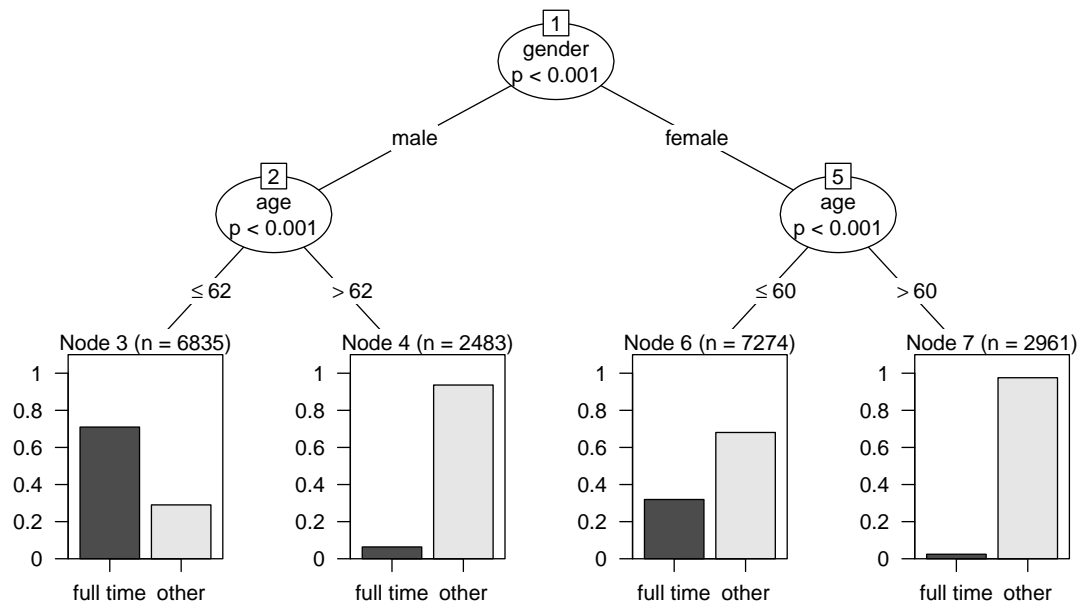


Figure 3.1: Classification tree: Simulated data indicating frequencies of full-time jobs.

Regression tree

- Calculate a regression tree

```
> rt_obj <- ctree(take_job ~ gender + age + nation + marital,
>                 control = ctree_control(minsplit = 10), data = dat_unempl)
```

- Examine the resulting regression tree object

```
> rt_obj
```

Conditional inference tree with 4 terminal nodes

Response: take_job

Inputs: gender, age, nation, marital

Number of observations: 950

1) gender == {male}; criterion = 1, statistic = 115.915

2) age <= 43; criterion = 0.988, statistic = 8.841

3)* weights = 236

4 SUPPLEMENT

```

2) age > 43
4)* weights = 147
1) gender == {female}
5) marital == {single}; criterion = 1, statistic = 49.76
6)* weights = 207
5) marital == {mar., mar.s, div., wid.}
7)* weights = 360

```

The same function `ctree()` is used to compute regression trees, where `take_job` is now a numerical response variable, namely the required income to take a job. According to the regression tree example (section 2.1) the covariates gender, age, nationality and marital status are used as potential splitting covariates.

- Graphical representation of the regression tree

```
> plot(rt_obj)
```

For regression trees the default representation uses box plots to display the distribution of the partitioned response variable in the end nodes.

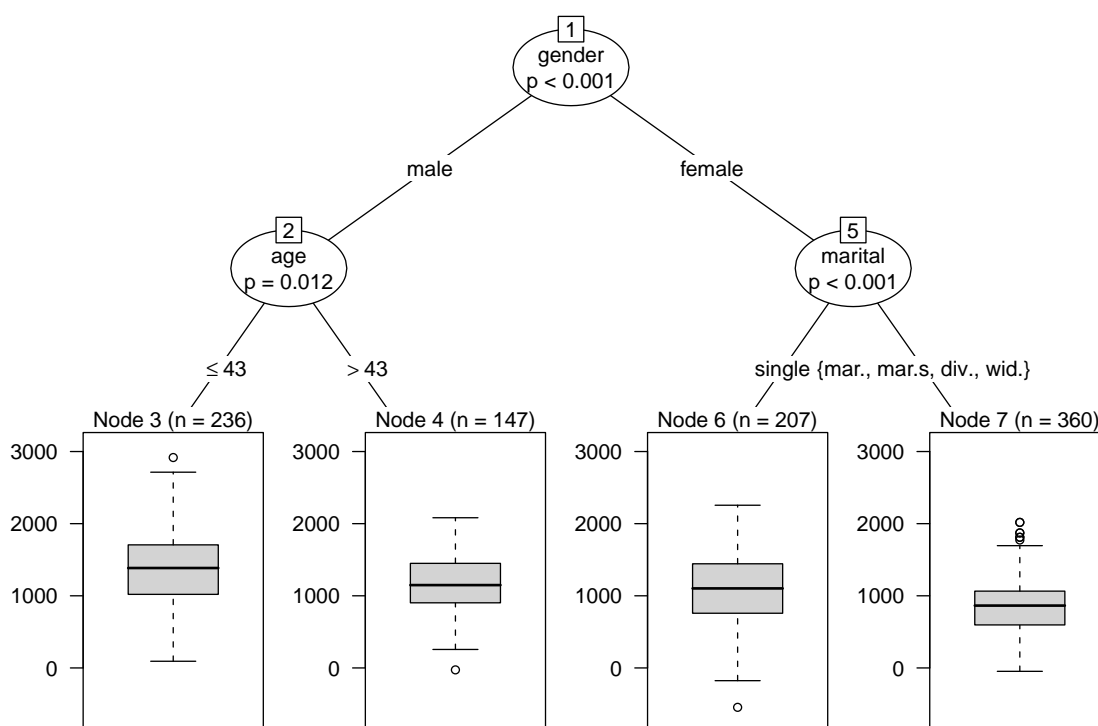


Figure 3.2: Regression tree: Simulated requested income for taking a job.

Model-based recursive partitioning

- Compute the recursive partitioning of a linear model

```
> mob_obj <- mob(jobvar ~ age + I(age^2) | gender + nation + marital,
>   control = mob_control(minsplit = 30), data = dat_job,
>   model = linearModel)
```

The function `mob()` is part of the `party` package as well (Zeileis et al. 2008). The structure of model-based recursive partitioning objects is more complex: Firstly, the model structure is specified. Here, a linear regression model is investigated (see Zeileis et al. 2008, for other specifications). Thus, the linear model explains the dependent variable `jobvar` through the independent variables `age + age2` and a u-shaped relationship between the requested income and the predictor variable `age` is assumed. In R code, the quadratic term for `age` is generated if `I(age^2)` is included in the formula (arithmetic operations have a different meaning in the formula context and the interpretation is inhibited using `I()`). After a vertical bar, potential splitting variables are included, such as `gender + nation + marital` in the example. Here the control argument is `control = mob_control(minsplit = 30, verbose=TRUE)`, allowing, e.g., to specify minimum splitting node sample sizes or to print test statistics during the computation process via `verbose=TRUE`.

- Print results from the model-based object

```
> temp <- coef(mob_obj)
> colnames(temp) <- c("Intercept", "age", "age sq.")
> printCoefmat(temp)
```

	Intercept	age	age sq.
2	998.916	22.613	-0.3640
4	748.667	11.673	-0.1204
5	1229.166	-17.144	0.1808

Tables 3.1 and 3.2 in the chapter are generated using the `printCoefmat()` function to inspect the estimated model parameters in the end nodes. The column names are combined into a vector using `c()` (starting with `c` for “concatenate”).

- If L^AT_EX is used for text setting, the library `xtable` can be used to generate a latex table (`xtable` has to be installed previous to first usage)

```
> library("xtable")
> print(xtable(temp, align="lrrr", digits=c(0,4,4,4),
>   caption="Coefficients of the models in the end nodes.",label="t1"),
>   type = "latex", sanitize.text.function = function(x){x})
```

	Intercept	age	age sq.
2	998.9155	22.6134	-0.3640
4	748.6667	11.6731	-0.1204
5	1229.1658	-17.1437	0.1808

Table 3.1: Coefficients of the linear models in the end nodes.

- Plot the model-based recursive partitioning object

```
> plot(mob_obj, tp_args = list(which = "age"), tnex = 2,
>      terminal_panel = node_bivplot(mob_obj, which = "age"))
```

The specification used here generates a scatter plot in the end nodes with the dependent variable `jobtime` and the independent variable `age` on the axes. In the chapter this plot occurs in Figure 3.3.

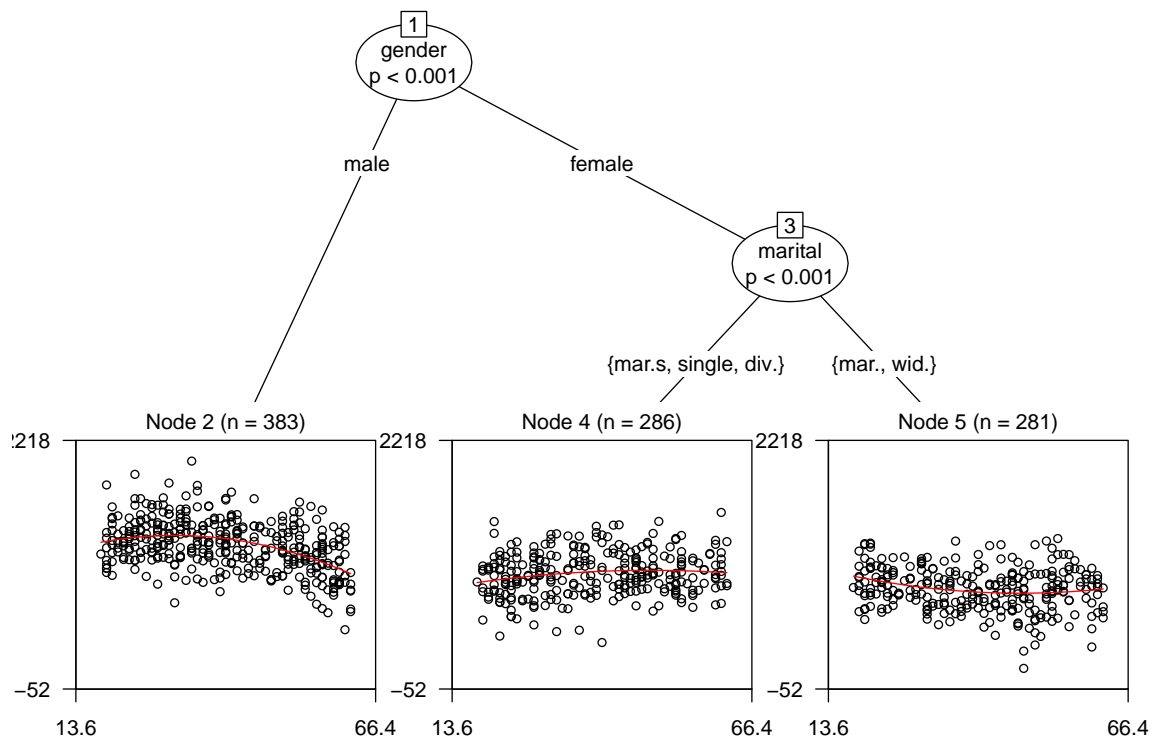


Figure 3.3: Model-based recursive partitioning: Simulated relationship between age and requested income.

Literature

Dalgaard, P. (2002). *Introductory Statistics with R*. New York: Springer.

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15, 651–674.

Leisch, F. (2002). Sweave: Dynamic generation of statistical reports using literate data analysis. In W. Härdle & B. Rönz (Eds.), *Compstat 2002 - Proceedings in computational statistics*, Heidelberg: Physica Verlag, (575–580).

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Zeileis, A., Hothorn, T., & Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17, 492–514.