

Software paper for submission to the Journal of Open Research Software

(1) Overview

Title

model4you: An R package for personalised treatment effect estimation

Paper Authors

1. Seibold, Heidi; 2. Zeileis, Achim; 3. Hothorn, Torsten

Paper Author Roles and Affiliations

1. [Postdoc; Institute for Medical Information Processing, Biometry, and Epidemiology, LMU Munich](#)
2. Professor; Department of Statistics, Faculty of Economics and Statistics, University of Innsbruck
3. Professor; Biostatistics Department, Epidemiology, Biostatistics & Prevention Institute, University of Zurich

Abstract

Typical models estimating treatment effects assume that the treatment effect is the same for all individuals. Model-based recursive partitioning allows to relax this assumption and to estimate stratified treatment effects (model-based trees) or even personalised treatment effects (model-based forests). With model-based trees one can compute treatment effects for different strata of individuals. The strata are found in a data-driven fashion and depend on characteristics of the individuals. Model-based random forests allow for a similarity estimation between individuals in terms of model parameters (e.g. intercept and treatment effect). The similarity measure can then be used to estimate personalised models. The R package *model4you* implements these stratified and personalised models [in the setting with two randomly assigned treatments](#) with a focus on ease of use and interpretability so that clinicians and other users can take the model they usually use for the estimation of the average treatment effect and with a few lines of code get a visualisation that is easy to understand and interpret.

Keywords

personalised medicine; subgroup analysis; model-based recursive partitioning; unbiased trees; treatment effect; random forest

Introduction

Studies in various fields randomly assign individuals to one of two groups with different exposure and then measure a response. For example, in clinical trials patients are assigned to one of two treatment groups where usually one treatment group receives a new treatment or drug and the other treatment group receives the standard of care or a placebo. Other examples are in A-B testing in marketing studies or any other two group comparisons such as the mathematics exam discussed

below, where students were divided into different exam groups and received slightly different exam tasks. In the following we will refer to the two groups as *treatment groups* and to the group indicator as *treatment indicator*, which always takes values 0 (individual in first group) and 1 (individual in second group).

Treatment effect estimation is often done using simple models with the binary treatment indicator as only covariate. In the example of a clinical trial the treatment indicator would be 1 if the patient receives the new treatment and 0 if the patient receives standard of care. In R such a simple model can be estimated as follows:

```
base_model <- model(response ~ treatment, data)
```

with `response` being the response measured, `treatment` being the treatment indicator and `data` being the data set containing these variables. The function `model()` can be replaced for example by `lm()` to estimate a linear model, `glm()` to estimate a generalised linear model or `survreg()` to estimate a parametric survival model. These models estimate intercept and treatment effect for all individuals in the data and allow for predicting the response of other individuals given they do or do not receive the treatment of interest.

For cases where the assumption that all individuals have the same intercept and treatment effect is too strict the R package *model4you* offers two options:

1. Model-based trees identify subgroups where within the subgroups the model parameters are similar and between groups the model parameters are different. This is achieved by finding instabilities in the model parameters with respect to a variable (characteristic) and recursively partitioning the data into groups. If, for example the algorithm finds that men and women have differing treatment effects, the data are partitioned into two subgroups. Details on model-based trees in general can be found in [Zeileis et al., 2008] and for the special use case for stratified treatment effect estimation in [Seibold et al., 2016]. Just a single line of code lets the user compute a model-based tree in R:

```
strat_models <- pmtree(base_model)
```

Note that `pmtree()` uses the data given in the call of the base model. It automatically uses variables not used in the model formula (in the example above `response` `treatment`) as potential subgroup defining variables. This can be edited using the `zformula` argument.

2. Personalised models use model-based random forests to estimate similarity of individuals in terms of model parameters. For each individual a personalised model can be estimated based on a weighted set of the original data, where the similarity measure corresponds to the weight. Details on the personalised models can be found in [Seibold et al., 2017]. Computing personalised models for all observations in the training data is simple:

```
pm_forest <- pmforest(base_model)
pers_models <- pmodel(pm_forest)
```

Again here the potential effect-modifying variables are taken by default as all variables not given in the model formula and can be defined using the `zformula` argument in `pmforest()`.

In the following we will present an example application for model-based trees and personalised models. For this we need to load the package and – to ensure reproducibility – set a random seed. Also for visualisations we need packages *ggplot2* [Wickham, 2016], *ggbeeswarm* [Clarke and Sherrill-Mix, 2017] and *gridExtra* [Auguie, 2017]. The data used is available in the *psychotools* package [Zeileis et al., 2018].

```
library("model4you")
set.seed(2017)

library("ggplot2")
theme_set(theme_classic())
library("ggbeeswarm")
library("gridExtra")
```

Mathematics exam analysis: In 2014 first-year business and economics students at the University of Innsbruck were divided into two examination groups. Group 1 took the exam in the morning and group 2 started after the first group finished. The exams for the two groups were slightly different. The data can be accessed and prepared as follows:

```
data("MathExam14W", package = "psychotools")

## scale points achieved to [0, 100] percent
MathExam14W$tests <- 100 * MathExam14W$tests/26
MathExam14W$pcorrect <- 100 * MathExam14W$nsolved/13

## select variables to be used
MathExam <- MathExam14W[, c("pcorrect", "group", "tests", "study",
                           "attempt", "semester", "gender")]
```

To investigate the correlation between exam group and exam performance, we compute a simple linear model regressing the percentage points of correct answers on the exam group.

```
bmod_math <- lm(pcorrect ~ group, data = MathExam)
```

The estimates and confidence intervals of this model can be computed via

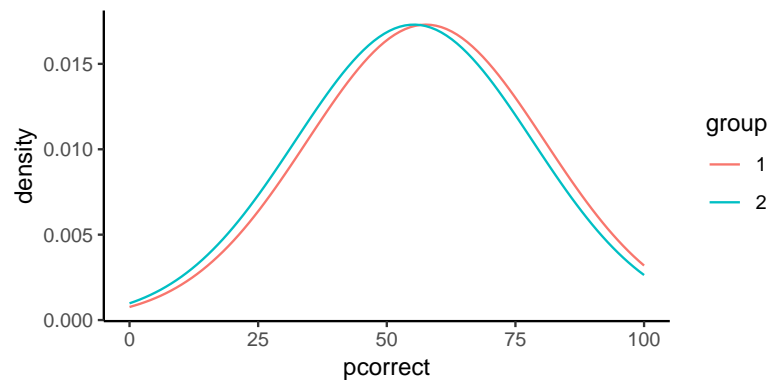


Figure 1: Density estimates of base model for the Mathematics Exam data.

```
cbind(estimate = coef(bmod_math), confint(bmod_math))

##           estimate      2.5 %   97.5 %
## (Intercept) 57.600184 55.122708 60.07766
## group2      -2.332414 -5.698108  1.03328
```

The model can be visualised by plotting the estimated densities (see Figure 1):

```
lm_plot(bmod_math)
```

Both the estimates and confidence intervals and the density curves suggest that there is almost no difference between the two groups. But does this really hold for all types of students?

A tree based on this model can be computed and visualised in only two lines of code:

```
tr_math <- pmtree(bmod_math, control = ctree_control(maxdepth = 2))
plot(tr_math, terminal_panel = node_pmtree(tr_math,
                                           plotfun = lm_plot))
```

We restrict the depth of the tree to two (`maxdepth = 2`) for illustration purposes. If this setting is not used a more complex tree would be estimated. Also the cutpoints and effect estimates are rounded in the plot. To view the tree in the R console, the `print()` function can be used.

```
print(tr_math)

## [1] root
## |   [2] tests <= 84.61538
## |   |   [3] tests <= 57.69231: n = 121
## |   |   (Intercept)      group2
## |   |   40.694789      -4.580056
```

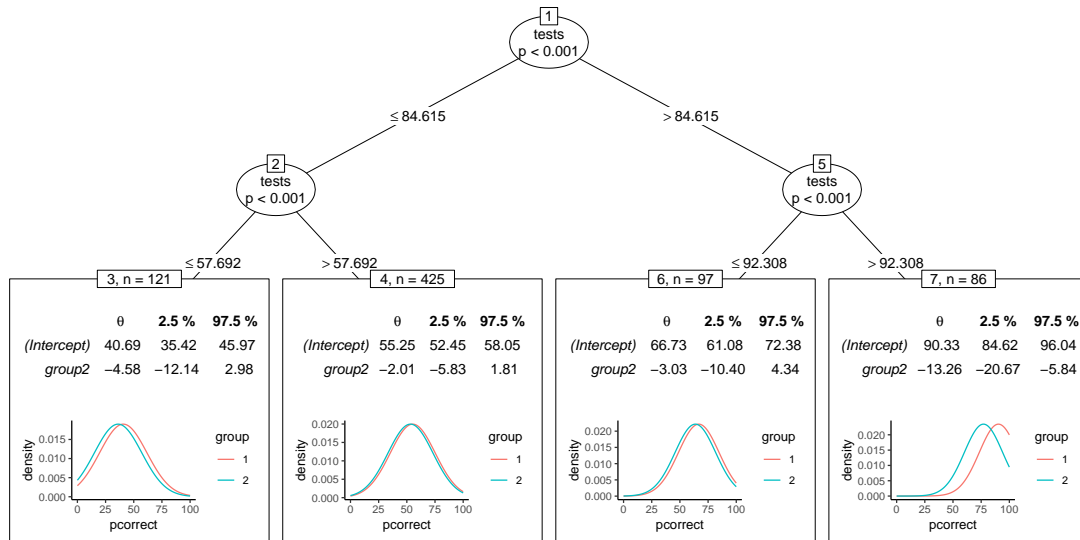


Figure 2: Personalised model tree for the Mathematics Exam datam.

```
## | | [4] tests > 57.69231: n = 425
## | | (Intercept) group2
## | | 55.251855 -2.012988
## | [5] tests > 84.61538
## | | [6] tests <= 92.30769: n = 97
## | | (Intercept) group2
## | | 66.730769 -3.033063
## | | [7] tests > 92.30769: n = 86
## | | (Intercept) group2
## | | 90.32967 -13.25576
##
## Number of inner nodes: 3
## Number of terminal nodes: 4
## Number of parameters per node: 2
## Objective function: -276167.6
```

The tree (see Figure 2) divides students based on the percentage of successful online tests. These online tests were conducted biweekly throughout the semester. The largest difference between the two exam groups is in the students who did very well in the online tests (more than 92.3 percent correct). The tree thus gives us much more information on the fairness of the exam than the simple linear model, which is that it does not seem to be fair for students who did very well throughout the semester (at this point we should state that the students self selected into the two exam groups which might also be the reason for differences in exam performance). Note that confidence intervals shown in the tree are not adjusted for selection of the cutpoints in the tree and should hence be interpreted as a measure for variability and not be used for inference.

Estimating personalised models is almost as simple as the stratified models:

```
forest_math <- pmforest(bmod_math)
pmods_math <- pmodel(forest_math)

## model parameters of first 6 students
head(pmods_math)

##      (Intercept)      group2
## 1      54.07712 -10.044714
## 2      40.17963  -6.326631
## 3      49.95619  -6.782954
## 4      53.19826  -9.367226
## 5      64.00692  -4.593500
## 6      41.13722  -6.465630
```

Dependence plots with the group effect (treatment effect) on the y-axis and the student characteristics on the x-axis are a good way of visualising the personalised models and for getting knowledge about the interactions between student characteristics and the exam group. Note that the plot is related to but not the same as the classical partial dependence plot [Friedman, 2001], as each point in the figure is one patient in the given data set. Since the percentage of successful online tests is measured on a grid, a beeswarm plot (a variant of jittered scatter plot) possibly shows the relationships even better than the scatter plot (both shown in Figure 3, Figure 3a includes a loess curve). We see a nonlinear relation between the percentage of successful online tests and the exam group effect. Students with great performance during the semester are estimated to have the strongest effect.

```
dpdat_math <- cbind(pmods_math, MathExam)

ggplot(dpdat_math, aes(x = tests, y = group2)) +
  geom_point(alpha = 0.2, size = 1) +
  geom_smooth(fill = NA, method = "loess") +
  theme(legend.position = "none") +
  ylab("estimated individual\nexam group effect")

ggplot(dpdat_math, aes(x = tests, y = group2, color = tests)) +
  geom_quasirandom(alpha = 0.5, size = 1) +
  theme(legend.position = "none") +
  ylab("estimated individual\nexam group effect")
```

For categorical variables such as the number of previous attempts to pass the exam (attempt) and gender we can use box plots, beeswarm plots or a combination thereof (as in Figure 4a). Figure 4b shows the number of previous attempts on the x-axis and the color indicates the gender of each student. This way we can see that for students writing the exam for the first time, gender seems to play a bigger role in terms of estimated individual exam group effect than for others.

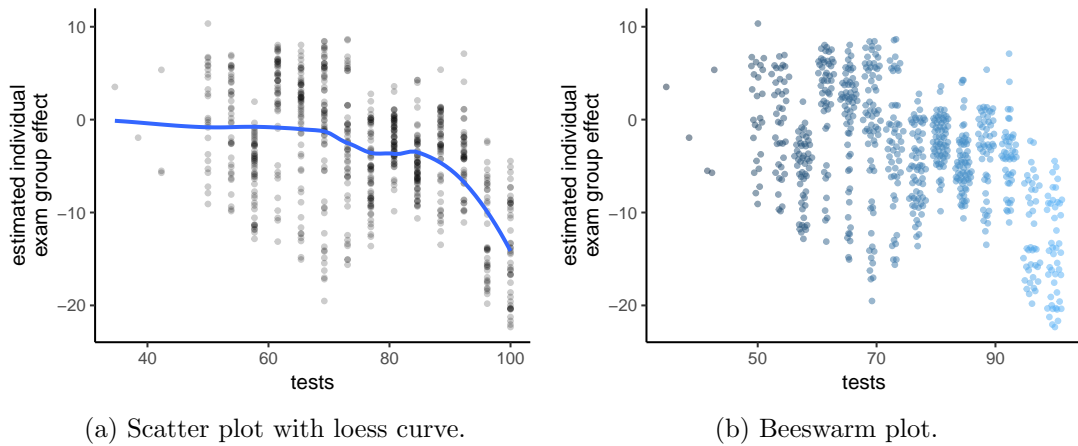


Figure 3: Dependence plot for percentage of tests successfully solved.

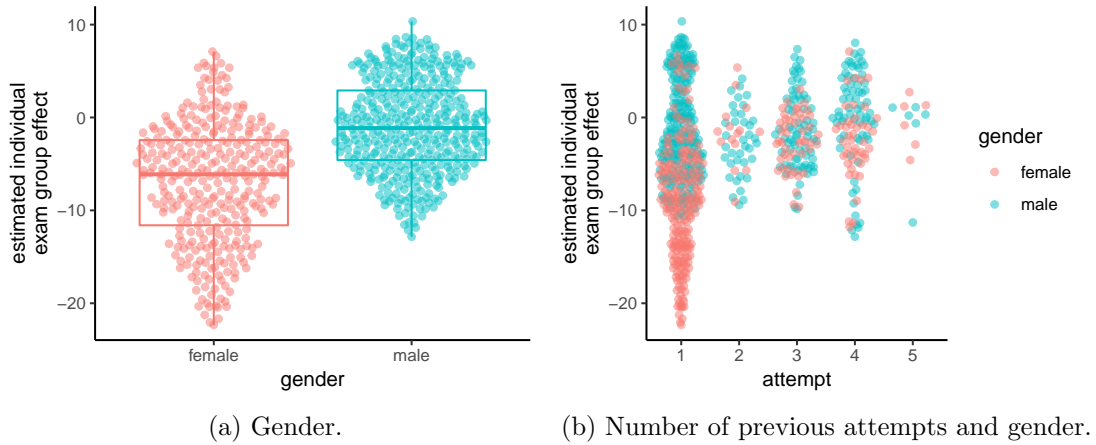


Figure 4: Dependence plots for the number of previous attempts and gender.

```
ggplot(dpdat_math, aes(x = gender, y = group2, color = gender)) +
  geom_boxplot() +
  geom_quasirandom(alpha = 0.5) +
  theme(legend.position = "none") +
  ylab("estimated individual\nexam group effect")

ggplot(dpdat_math, aes(x = attempt, y = group2, color = gender)) +
  geom_quasirandom(alpha = 0.5) +
  ylab("estimated individual\nexam group effect")
```

With the tools provided by the *model4you* package it is very simple to create understandable stratified and personalised models and compelling visualisations that can be used to communicate these models.

Implementation and architecture

The R package *model4you* is focused on ease of use and interpretability. Users can take a simple model that they know and understand as basis. Models currently available for use are linear models (function `lm()` in R), generalised linear models (`glm()`) and survival models (`survreg()` and `coxph()` from the R package *survival*). Models are restricted to those with a single binary covariate (e.g. two treatment options). The user can simply plug the model object into `pmtree()` or `pmforest()` depending on whether they want subgroup wise or personalised models.

The basis for these functionalities is provided by the *partykit* package which is a widely used R package for trees and forests [Hothorn and Zeileis, 2015, 2017]. The *model4you* package provides wrappers for the well implemented and tested functions `partykit::ctree()` and `partykit::cforest()` and extends the functionalities to allow for the computation of personalised models and to improve usability and interpretability.

The *partykit* package provides the basis for functionalities in other packages namely *glmertree*, *psychotree*, *betareg*, *disttree*, *lagsarlm tree*, *palm tree* (all on CRAN), and *trtf* (currently available on R-Forge).

Quality control

All packages on CRAN undergo standard checks for compatibility with the R package ecosystem. The R package contains examples and tests. These were run and checked on Linux 86_64 and Windows.

(2) Availability**Operating system**

Should work on all operating systems that run R.

Programming language

R (version 3.1.0 or higher)

Additional system requirements

None.

Dependencies

R, *partykit* package (version 1.2 or higher)

List of contributors

Same as the authors: Heidi Seibold, Achim Zeileis and Torsten Hothorn

Software location:**Archive**

Name: CRAN

Persistent identifier: <https://cran.r-project.org/package=model4you>

Licence: GPL-2 | GPL-3

Publisher: Heidi Seibold

Version published: The version number of the software archived.

Date published: dd/mm/yy

Code repository

Name: R-forge

Persistent identifier: <https://r-forge.r-project.org/projects/partykit/>

Licence: GPL-2 | GPL-3

Date published: dd/mm/yy

Language

English

(3) Reuse potential

The software is intentionally written to make usage as simple as possible. The most prominent use case are clinical trials where the assumption of an average treatment effect for all patients is too strict and the efficacy of the treatment depends on patient characteristics (e.g. gender, biomarkers, etc.). For subgroup analyses (stratified treatment effects) model-based trees (`pmtree()`) can be used; For personalised treatment effects model-based forests (`pmforest()`) provide a way of estimating similarity between patients and using this similarity measure to estimate personalised models (`pmodel()`). The target audience are people who deal with heterogeneous treatment effects, such as medical researchers, pharmaceutical companies or analysts in marketing (A-B testing). In general the software is useful to researchers dealing with scenarios where two exposures are compared and responses of subjects possibly depend on other variables. [Currently the packages supports a limited set of model types \(linear and generalised linear models and survival models\). Further models can be added.](#)

We encourage users to use the party tag on Stackoverflow (<http://stackoverflow.com/questions/tagged/party>) in case of questions or problems.

Acknowledgements

Funding statement

Heidi Seibold and Torsten Hothorn were financially supported by the Swiss National Science Foundation (grants 205321_163456 and IZSEZ0_177091).

Competing interests

The authors declare that they have no competing interests.

References

Baptiste Auguie. *gridExtra: Miscellaneous Functions for "Grid" Graphics*, 2017. URL <https://CRAN.R-project.org/package=gridExtra>. R package version 2.3.

Erik Clarke and Scott Sherrill-Mix. *ggbeeswarm: Categorical Scatter (Violin Point) Plots*, 2017. URL <https://CRAN.R-project.org/package=ggbeeswarm>. R package version 0.6.0.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. URL <http://www.jstor.org/stable/2699986>.

- Torsten Hothorn and Achim Zeileis. `partykit`: A modular toolkit for recursive partytioning in R. *Journal of Machine Learning Research*, 16:3905–3909, 2015. URL <http://jmlr.org/papers/v16/hothorn15a.html>.
- Torsten Hothorn and Achim Zeileis. *partykit: A Toolkit for Recursive Partytioning*, 2017. URL <http://CRAN.R-project.org/package=partykit>. R package version 1.2-0.
- Heidi Seibold, Achim Zeileis, and Torsten Hothorn. Model-based recursive partitioning for subgroup analyses. *International Journal of Biostatistics*, 12(1):45–63, 2016. doi: 10.1515/ijb-2015-0032.
- Heidi Seibold, Achim Zeileis, and Torsten Hothorn. Individual treatment effect prediction for amyotrophic lateral sclerosis patients. *Statistical Methods in Medical Research*, 2017. doi: 10.1177/0962280217693034. Online first.
- Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL <http://ggplot2.org>.
- Achim Zeileis, Torsten Hothorn, and Kurt Hornik. Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2):492–514, 2008. doi: 10.1198/106186008X319331.
- Achim Zeileis, Carolin Strobl, Florian Wickelmaier, Basil Komboz, and Julia Kopf. *psychotools: Infrastructure for Psychometric Modeling*, 2018. URL <https://CRAN.R-project.org/package=psychotools>. R package version 0.4-3.

Copyright Notice

Authors who publish with this journal agree to the following terms:

Authors retain copyright and grant the journal right of first publication with the work simultaneously licensed under a Creative Commons Attribution License that allows others to share the work with an acknowledgement of the work’s authorship and initial publication in this journal.

Authors are able to enter into separate, additional contractual arrangements for the non-exclusive distribution of the journal’s published version of the work (e.g., post it to an institutional repository or publish it in a book), with an acknowledgement of its initial publication in this journal.

By submitting this paper you agree to the terms of this Copyright Notice, which will apply to this submission if and when it is published by this journal.