An Introduction to the GPLTR package

Cyprien Mbogning
Inserm UNIT 669, France

April 22, 2014

Contents

1 Introduction

This document is intended to give a short overview of the methods found in the `GPLTR` package. The acronym `GPLTR` is designed for — Generalized Partially Linear Tree-based Regression model —.

The `GPLTR` programs build classification or regression models of a very general structure using a three stage procedure with several additional tools; the resulting model is an hybrid model combining a generalized linear model with an additional tree part on the same scale. The model was first proposed by Chen et al.[2] for genetic epidemiology study studies in order to assess complex joint gene-gene and gene-environment effects taking into account confounding variables. In practice, the GPLTR models represent a new class of semi-parametric regression models that integrates the advantages of generalized linear regression and tree-structure models. To our best knowledge, there is currently no implemented package dealing with this kind of model. The available classical tree-based methods do not provide a way for controlling confounding factors outside the tree part (the final tree is generally a mixture of confounders and explanatory variables lacking of clear interpretation and resulting in a distorted joint effect).

2 GPLTR model

Denote $\mathbf{Y}$ the outcome of interest, $\mathbf{X}$ a set of confounding variables, and $\mathbf{G}$ the explanatory variables. The model fitted inside the `GPLTR` package is specified by:

$$g\left(\mathbb{E}\left(\mathbf{Y}|\mathbf{X},\mathbf{G}\right)\right) = \mathbf{X}'\theta + \beta_T F\left(T\left(\mathbf{G}\right)\right), \tag{1}$$

where $g(\cdot)$ is a known link function (generalized linear model), $F\left(T\left(\mathbf{Z}\right)\right)$ is a vector of indicator variables representing the leaves of the tree $T\left(\mathbf{G}\right)$.

The variables considered in the linear part (confounding variables or variables we wish to control) of the model 1 have a direct impact on the structure of the tree, beginning by the split criterion and ending by the pruning procedure.

3 Fitting methods

The method we used in this package to fit the model 1 can be summarized into three major steps:

Step1 Fit the linear part and build a maximal tree within an iterative procedure by playing on several offsets terms. The nodes of the tree are splitted by maximizing a deviance criterion, while an intercept coefficient is fitted inside the node using the corresponding glm with the linear part considered as offset.

Step2 In order to prune back the maximal tree obtained previously, we use a forward procedure to build a sequence of nested subtrees.

Step3 The optimal tree is selected, using either a BIC criterion, a AIC criterion, a K-fold Cross-validation procedure on the underlying GPLTR models corresponding to the nested trees sequence. The original parametric bootstrap test procedure proposed by Chen et al. is also available.

We further propose a procedure to test the joint effect of the selected tree while adjusting for confounders. The users are encouraged to read the recent paper of Mbogning et al. [5] for a more thorough explanation about the model and the methods.

4 Illustration via several examples

In the following, we will present the results obtained on the publicly available ”burn” Data Set (Times to Infection for Burn Patients from the book of Klein and Moeschberger [4]). This dataset comes from a study (Ichida et al.[3]) that evaluates a protocol change in disinfectant practices for a cohort of 154 patients. A complete description of the data is also available inside the GPLTRpackage documentation.

In this example, the dependent variable is the administration of prophylactic antibiotic treatment ($D2$: yes/no), the confounding variable is the gender ($Z2$: male/female) and the potential explanatory variables are: ethnicity, severity of the burn as measured by percentage of total surface area of body burned, burn site (head, buttocks, trunk, upper legs, lower legs, respiratory tract), and type of burn (chemical, scald, electric, flame). In this analysis, we included gender as a confounding factor since this factor has already been described as related to infections among burn patients (Wisplinghoff et al. [7]). Such adjustment for confounders cannot be performed within the classical CART framework.

Results obtained with the classical CART algorithm

First of all, we have fitted a classical tree model on the dependent variable $D2$, using the CART algorithm [1] via the ’rpart’ routines [6] of the R software:

```
> data(burn)
> cfit <- rpart(D2 ~ Z1  + Z2 + Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9
                    + Z10 + Z11, data = burn, method = "class")
> par(mar = rep(0.1, 4))
> plot(cfit, uniform = TRUE)
>  text(cfit, xpd = TRUE, use.n = TRUE)
```

Figure (1) represents the tree obtained using the classical CART algorithm.

Results obtained with our proposed method within the GPLTR package

A logistic partially linear tree-based regression model is fitted by using our proposed method:

```
> #setting the parameters
> args.rpart <- list(minbucket = 10, maxdepth = 4, cp = 0)
> family <- "binomial"
> X.names = "Z2"
> Y.name = "D2"
> G.names = c('Z1','Z3','Z4','Z5','Z6','Z7','Z8','Z9','Z10','Z11')
> # Build the tree with an adjustment  on gender (Z2)
```

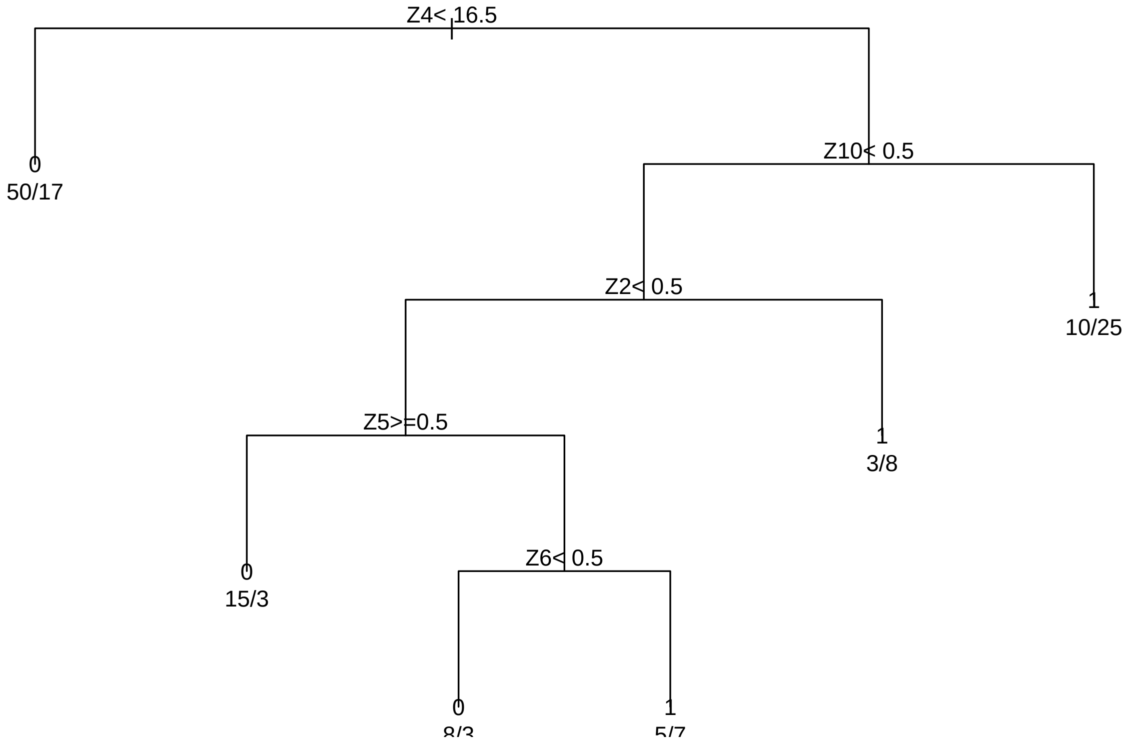

Figure 1: Tree obtained on the burn dataset with rpart.

```
> fit_pltr <- pltr.glm(burn, Y.name, X.names, G.names, args.rpart =
                        args.rpart, family = family,iterMax =8, iterMin = 6,
                          verbose = TRUE)
Iteration process...

Iteration  1 in PLTR; Diff_norm_gamma =  0.3450989
Iteration  2 in PLTR; Diff_norm_gamma =  0.1035102
Iteration  3 in PLTR; Diff_norm_gamma =  0.1555209
Iteration  4 in PLTR; Diff_norm_gamma =  0.04005871
Iteration  5 in PLTR; Diff_norm_gamma =  0.01202298
Iteration  6 in PLTR; Diff_norm_gamma =  0.003612611
Iteration  7 in PLTR; Diff_norm_gamma =  0.001085865
Iteration  8 in PLTR; Diff_norm_gamma =  0.0003264183
End of iteration process
Number of iterations:  9
> tree_select <- best.tree.BIC.AIC(xtree = fit_pltr$tree,burn,Y.name,
                          X.names, G.names, family = family, verbose = FALSE)
> summary(tree_select$tree$BIC)
Call:
rpart(formula = as.formula(paste(Y.name, " ~ ", paste(G.names,
    collapse = " + "), paste("+ offset(offsetX)"))), data = eval(parse(text = paste("dat
    product, "hat_gamma)"))), method = method, control = args.rpart)
  n= 154

            CP nsplit rel error
1 0.0604978419      0 1.0000000
2 0.0408578707      1 0.9395022
3 0.0318009580      2 0.8986443
4 0.0261741219      3 0.8668433
5 0.0147618258      5 0.8144951
6 0.0073443205      6 0.7997333
7 0.0025318226      7 0.7923889
8 0.0002897679      8 0.7898571
9 0.0000000000      9 0.7895673

Variable importance
 Z4  Z1 Z10  Z6  Z5 Z11  Z7  Z8  Z9
 32  26  16   9   5   5   4   2   1
```

```
Node number 1: 154 observations,    complexity param=0.06049784
events = 63, coef = -0.6005625, deviance = 202.69720
  left son=2 (64 obs) right son=3 (90 obs)
  Primary splits:
      Z4  < 15.5 to the left,  improve=12.262740, (0 missing)
      Z6  < 0.5  to the left,  improve= 7.487501, (0 missing)
      Z1  < 0.5  to the left,  improve= 6.930854, (0 missing)
      Z10 < 0.5  to the left,  improve= 6.749544, (0 missing)
      Z8  < 0.5  to the left,  improve= 5.227568, (0 missing)
  Surrogate splits:
      Z7  < 0.5  to the left,  agree=0.649, adj=0.156, (0 split)
      Z11 < 3.5  to the left,  agree=0.636, adj=0.125, (0 split)
      Z5  < 0.5  to the left,  agree=0.610, adj=0.063, (0 split)
      Z8  < 0.5  to the left,  agree=0.604, adj=0.047, (0 split)

Node number 2: 64 observations,    complexity param=0.03180096
events = 16, coef = -1.3621820, deviance =  71.21259
  left son=4 (25 obs) right son=5 (39 obs)
  Primary splits:
      Z1  < 0.5  to the left,  improve=6.4459640, (0 missing)
      Z10 < 0.5  to the right, improve=2.0880440, (0 missing)
      Z4  < 9.5  to the right, improve=0.9623458, (0 missing)
      Z5  < 0.5  to the right, improve=0.7410280, (0 missing)
      Z11 < 3.5  to the right, improve=0.4042443, (0 missing)
  Surrogate splits:
      Z10 < 0.5  to the right, agree=0.672, adj=0.16, (0 split)
      Z4  < 14.5 to the right, agree=0.656, adj=0.12, (0 split)
      Z7  < 0.5  to the left,  agree=0.625, adj=0.04, (0 split)
      Z11 < 2.5  to the left,  agree=0.625, adj=0.04, (0 split)

Node number 3: 90 observations,    complexity param=0.04085787
events = 47, coef = -0.1251768, deviance = 119.22180
  left son=6 (55 obs) right son=7 (35 obs)
  Primary splits:
      Z10 < 0.5  to the left,  improve=8.281774, (0 missing)
      Z6  < 0.5  to the left,  improve=4.774796, (0 missing)
      Z1  < 0.5  to the left,  improve=4.628120, (0 missing)
      Z4  < 39   to the left,  improve=3.877164, (0 missing)
      Z8  < 0.5  to the left,  improve=2.295146, (0 missing)
  Surrogate splits:
      Z4 < 47.5 to the left,  agree=0.711, adj=0.257, (0 split)
      Z5 < 0.5  to the left,  agree=0.633, adj=0.057, (0 split)

Node number 4: 25 observations
events = 2, coef = -2.6717880, deviance =  14.89458

Node number 5: 39 observations
events = 14, coef = -0.8618721, deviance =  49.87204

Node number 6: 55 observations,    complexity param=0.02617412
events = 22, coef = -0.6175948, deviance =  68.93242
  left son=12 (37 obs) right son=13 (18 obs)
  Primary splits:
      Z6 < 0.5  to the left,  improve=4.886466, (0 missing)
      Z4 < 39   to the left,  improve=4.504188, (0 missing)
      Z8 < 0.5  to the left,  improve=2.328673, (0 missing)
      Z1 < 0.5  to the left,  improve=2.056546, (0 missing)
      Z5 < 0.5  to the right, improve=1.881947, (0 missing)
  Surrogate splits:
      Z4  < 36.5 to the left,  agree=0.782, adj=0.333, (0 split)
      Z11 < 2.5  to the right, agree=0.727, adj=0.167, (0 split)
      Z7  < 0.5  to the right, agree=0.691, adj=0.056, (0 split)
      Z9  < 0.5  to the left,  agree=0.691, adj=0.056, (0 split)

Node number 7: 35 observations
events = 25, coef =  0.6995323, deviance =  42.00763

Node number 12: 37 observations,    complexity param=0.02617412
events = 11, coef = -1.0794600, deviance =  42.43535
  left son=24 (20 obs) right son=25 (17 obs)
  Primary splits:
      Z1 < 0.5  to the left,  improve=5.7243740, (0 missing)
      Z4 < 21.5 to the right, improve=2.7335920, (0 missing)
      Z8 < 0.5  to the left,  improve=0.3237422, (0 missing)
      Z5 < 0.5  to the right, improve=0.1542775, (0 missing)
  Surrogate splits:
      Z4 < 23.5 to the right, agree=0.649, adj=0.235, (0 split)
      Z9 < 0.5  to the left,  agree=0.568, adj=0.059, (0 split)

Node number 13: 18 observations
```

```
events = 11, coef =  0.2519882, deviance =  21.61060

Node number 24: 20 observations
events = 3, coef = -2.0596390, deviance =  14.90882

Node number 25: 17 observations
events = 8, coef = -0.2338515, deviance =  21.80216
> summary(tree_select$fit_glm$BIC)
Call:
glm(formula = xformula, family = family, data = xdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9830  -0.8274  -0.3593   0.9052   2.3547

Coefficients:
                                                                      Estimate
(Intercept)                                                             0.6805
Z2                                                                      1.1349
as.integer(Z4 < c(15.5) & Z1 < c(0.5))                                 -3.3883
as.integer(Z4 < c(15.5) & Z1 >= c(0.5))                                -1.5766
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 < c(0.5) & Z1 < c(0.5))   -2.7869
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 < c(0.5) & Z1 >= c(0.5))  -0.9263
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 >= c(0.5))                -0.4475
                                                                      Std. Error
(Intercept)                                                               0.3882
Z2                                                                        0.4686
as.integer(Z4 < c(15.5) & Z1 < c(0.5))                                    0.8405
as.integer(Z4 < c(15.5) & Z1 >= c(0.5))                                   0.5175
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 < c(0.5) & Z1 < c(0.5))      0.7534
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 < c(0.5) & Z1 >= c(0.5))     0.6241
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 >= c(0.5))                   0.6238
                                                                      z value
(Intercept)                                                             1.753
Z2                                                                      2.422
as.integer(Z4 < c(15.5) & Z1 < c(0.5))                                 -4.031
as.integer(Z4 < c(15.5) & Z1 >= c(0.5))                                -3.047
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 < c(0.5) & Z1 < c(0.5))   -3.699
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 < c(0.5) & Z1 >= c(0.5))  -1.484
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 >= c(0.5))                -0.717
                                                                      Pr(>|z|)
(Intercept)                                                           0.079594
Z2                                                                    0.015445
as.integer(Z4 < c(15.5) & Z1 < c(0.5))                                5.54e-05
as.integer(Z4 < c(15.5) & Z1 >= c(0.5))                               0.002315
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 < c(0.5) & Z1 < c(0.5))  0.000217
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 < c(0.5) & Z1 >= c(0.5)) 0.137707
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 >= c(0.5))               0.473180

(Intercept)                                                           .
Z2                                                                    *
as.integer(Z4 < c(15.5) & Z1 < c(0.5))                                ***
as.integer(Z4 < c(15.5) & Z1 >= c(0.5))                               **
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 < c(0.5) & Z1 < c(0.5))  ***
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 < c(0.5) & Z1 >= c(0.5))
as.integer(Z4 >= c(15.5) & Z10 < c(0.5) & Z6 >= c(0.5))
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 208.37  on 153  degrees of freedom
Residual deviance: 165.03  on 147  degrees of freedom
AIC: 179.03

Number of Fisher Scoring iterations: 5
> par(mfrow = c(1,2), mar = rep(0.1, 4))
> plot(fit_pltr$tree, uniform = TRUE, margin = 0.05)
> plot(tree_select$tree$BIC, uniform = TRUE, margin = 0.05)
> text(tree_select$tree$BIC, xpd = TRUE)
```

- The underlying method behind the ’binomial’ family above is a new one, different from those implemented inside the `rpart` package. The splitting criterion is based on a logistic deviance criterion considering the linear part as offset (See Mbogning et al.[5]).

- The child nodes of node x are always $2x$ and $2x + 1$, to help in navigating the tree summary (compare the summary to figure 2).

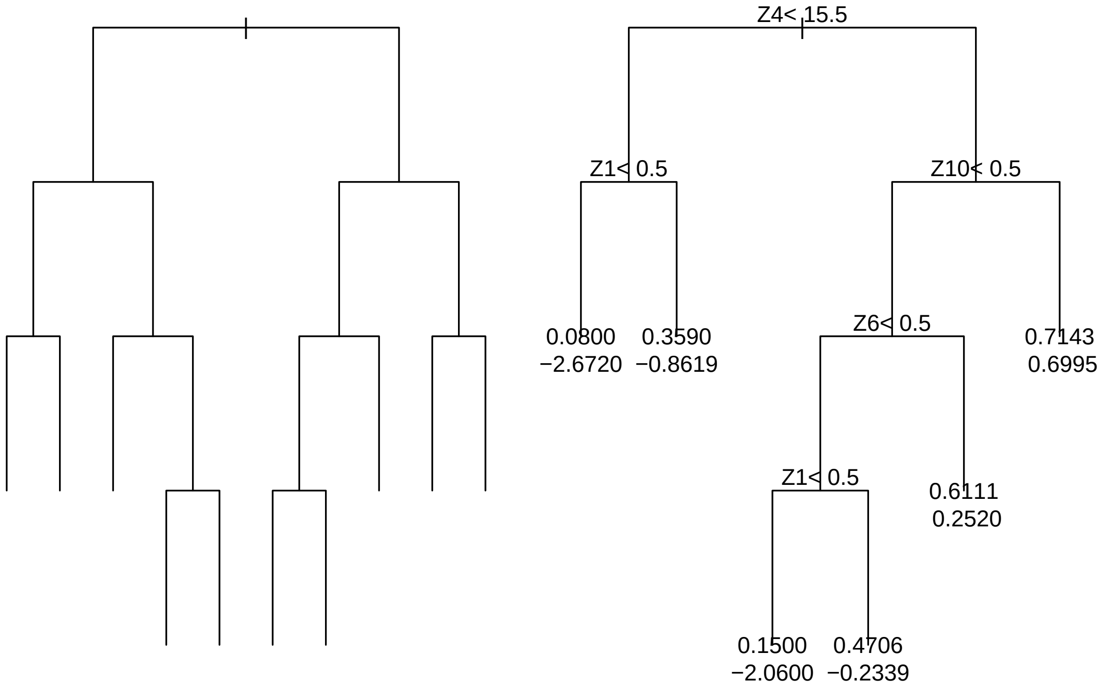

Figure 2: The figure on the left is the maximal tree obtained with pltr.glm; the figure on the right is a pruned tree via a BIC criterion

- They are many Items in the tree summary list:
  - the complexity table
  - the variable importance
  - the node number
  - the number of cases within the node
  - the number of events (number of cases with attribute 1) inside the node
  - the logistic intercept coefficient fitted inside the node, which represents the summary statistic of the node. This coefficient represents the predicted value of the node. that's the main difference with a conventional tree where the predicted value is the modal class of the node.
  - the logistic deviance of the previous model inside the node which is used as the splitting criterion
- * indicates that the node is terminal.
- the first split is on the Percentage of total surface area burned (Z_4). 64 individuals with $Z_4 < 15.5$ go to the left while the remaining 90 go to the right. The split with the maximum number of events is always on the right.
- The improvement listed is the change in deviance for the split, ie., $D(parent) - (D(leftson) + D(rightson))$, where D is the deviance operator. This is similar to a likelihood ratio test statistic.
- the other nodes can be described similarly.

For all the two models (tree with rpart (Fig 1) and the tree with our proposed procedure (Fig 2)), the ???rst split is due to the percentage of total surface area burned ($(< 16.5\%, > 16.5\%)$ for `rpart` and $(< 15.5\%, > 15.5\%)$ for the proposed method). The subsequent splits are di???erent between patients having a high or low percentage of total surface area burned. The classical rpart model shows only one split whereas our proposed PLTR model shows two splits. With the exception of CART model where no split occurs, the subsequent split for the group of patients with a low percentage of area burned is due to the initial treatment (routine bathing/body cleansing). For high percentage of surface area burned, the split for the two models is due to the respiratory tract damage. Our proposed procedure identi???es other splits due to the treatment and the buttock burns. In particular, we observed that the group of patients with a high percentage of surface area of body burned, without tract respiratory damage, buttock injury and without routine bathing shows a lower proportion of prophylactic antibiotics administration. This group is not detected by the original PLTR method. It is worth noting that the confounding factor $Z2$ is significant with a higher proportion of prophylactic antibiotics administration among women.

The particularity of the PLTR model is that in addition with the tree part, the final model is a classical logistic model with new risk factors emerging from the tree part. The summary of the final logistic model is presented within the R code above. We can see for example that the new risk factor constitute by individuals sharing the attributes $Z4 < 15.5$ and $Z1 < 0.5$ is higly significant. Similar interpretation can be made for other factors.

5 Compute the generalization error of the procedure

We can further compute the generalization error of the procedure. This can be computed via the function `best.tree.CV` which can also provide the best tree based on a K-fold cross-validation procedure.

```
> tree_selected <- best.tree.CV(fit_pltr$tree, burn, Y.name, X.names,
                    G.names, family = family, args.rpart = args.rpart,
                    epsi = 0.001, iterMax = 15, iterMin = 8, ncv = 10,
                    verbose = FALSE)
Max tree size  10 reached
Max tree size  10 reached
> tree_selected$CV_ERROR
[[1]]
[1] 0.3221053

[[2]]
[1] 0.3701754 0.3782456 0.3435088 0.3501754 0.3621053
[6] 0.3354386 0.3221053 0.3287719
> Bic_size <- sum(tree_select$tree$BIC$frame$var == '<leaf>')
> ## Bic_size <- tree_select$best_index[[1]]
>
> CV_ERROR_BIC <- tree_selected$CV_ERROR[[2]][Bic_size]
> CV_ERROR_BIC
[1] 0.3354386
```

6 Test the joint effect of the selected tree while adjusting for confounders.

We can also test the joint effect of the selected tree after adjusting for the confounding variable

```
> args.parallel = list(numWorkers = 1, type = "PSOCK")
> index = Bic_size
> # p_value <- p.val.tree(xtree = fit_pltr$tree, data_pltr, Y.name, X.names,
> #             G.names, B = 1000, args.rpart = args.rpart, epsi = 1e-3,
> #             iterMax = 15, iterMin = 8, family = family, LB = FALSE,
> #             args.parallel = args.parallel, index = index, verbose =
> #             FALSE)
> # p_value$P.value
```

References

[1] L. Breiman, J. H. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth International Group, Belmont, California, 1984.

[2] Jimbo Chen, Kai Yu, Ann Hsing, and Terry M. Therneau. A partially linear tree-based regression model for assessing complex joint gene-gene and gene-environment effects. *Genetic Epidemiology*, 31:238–251, 2007.

[3] J. M. Ichida, J. T. Wassell, M. D. Keller, and L. W. Ayers. Evaluation of Protocol Change in Burn-Care Management Using the Cox Proportional Hazards Model with Time-Dependent Covariates. *Statistics in Medicine*, 12:301–310, 1993.

[4] J. P. Klein and M. L. Moeschberger. *SURVIVAL ANALYSIS Techniques for Censored and Truncated Data.* Springer, New York, second edition, 2003.

[5] Cyprien Mbogning, Hervé Perdry, Wilson Toussile, and Philippe Broët. A novel tree-based procedure for deciphering the genomic spectrum of clinical disease entities. *Journal of Clinical Bioinformatics*, 4(6), 2014.

[6] Terry M. Therneau and Elizabeth J. Atkinson. An introduction to recursive partitioning using the RPART routines. *Mayo Foundation*, 2013.

[7] Hilmar Wisplinghoff, Walter Perbix, and Harald Seifert. Risk Factors for Nosocomial Bloodstream Infections Due to Acinetobacter baumannii: A Case-Control Study of Adult Burn Patients. *Clin. Infect. Dis.*, 28(1):59–66, 1999.