Package 'TSVC'

March 2, 2022

Type Package

Title Tree-Structured Modelling of Varying Coefficients

Version 1.2.2

Date 2022-03-02

Author Moritz Berger

Depends plotrix, mgcv

Suggests AER

Maintainer Moritz Berger <moritz.berger@imbie.uni-bonn.de>

Description

Fitting tree-structured varying coefficient models (Berger et al. (2019), <doi:10.1007/s11222-018-9804-8>). Simultaneous detection of covariates with varying coefficients and effect modifiers that induce varying coefficients if they are present.

License GPL-2

LazyLoad yes

RoxygenNote 7.1.2

NeedsCompilation no

Repository CRAN

Date/Publication 2022-03-02 17:00:02 UTC

R topics documented:

plot.TSVC *Plotting of Varying Coefficient Trees*

Description

Visualization of trees of effects of covariates that vary with the values of one or several effect modifiers.

Usage

```
## S3 method for class 'TSVC'
plot(
  x,
  variable,
  cex.lines = 2,
  cex.branches = 1,
  cex.coefs = 1,
  cex.main = 1,
  title = NULL,
  ...
)
```

Arguments

`x`	object of class `TSVC`.
`variable`	name of the variable, for which the tree shall be plotted.
`cex.lines`	width of branches of the tree.
`cex.branches`	size of the labels of the tree.
`cex.coefs`	size of the coefficients in the terminal nodes of the tree.
`cex.main`	size of the title of the tree.
`title`	optional title, which is addded to the tree; if `title=NULL` the title is the name of the variable in the data.
`...`	further arguments passed to or from other methods.

Author(s)

Moritz Berger <Moritz.Berger@imbie.uni-bonn.de>
`https://www.imbie.uni-bonn.de/personen/dr-moritz-berger/`

References

Berger, M., G. Tutz and M. Schmid (2019). Tree-Structured Modelling of Varying Coefficients. Statistics and Computing 29, 217-229, https://doi.org/10.1007/s11222-018-9804-8.

See Also

TSVC, predict.TSVC, summary.TSVC

Examples

```
# Swiss Labour Market
library(AER)
data("SwissLabor")

# recode factors
sl <- SwissLabor
sl$participation <- as.numeric(sl$participation)-1
sl$foreign       <- as.numeric(sl$foreign)-1

## Not run:
fit1 <- TSVC(participation~income+age, data=sl, family=binomial(link="logit"),
             nperm=1000, trace=TRUE)
plot(fit1, "income")

## End(Not run)
```

predict.TSVC *Prediction from Varying Coefficient Trees*

Description

Obtains predictions from a fitted TSVC object.

Usage

```
## S3 method for class 'TSVC'
predict(object, X_new = NULL, ...)
```

Arguments

object	a fitted object of class TSVC.
X_new	optionally, data frame of class data.frame which contains the variables with which to predict. If NULL, the fitted linear predictors are use.
...	further arguments passed to predict.glm.

Details

predict.TSVC is a wrapper function of predict.glm, which obtains predictions for objects of class glm. Further arguments can be passed to predict.glm via the '...'-argument.

Author(s)

Moritz Berger <moritz.berger@imbie.uni-bonn.de>
https://www.imbie.uni-bonn.de/personen/dr-moritz-berger/

References

Berger, M., G. Tutz and M. Schmid (2019). Tree-Structured Modelling of Varying Coefficients. Statistics and Computing 29, 217-229, https://doi.org/10.1007/s11222-018-9804-8.

See Also

TSVC, plot.TSVC, summary.TSVC

Examples

```
# Swiss Labour Market
library(AER)
data("SwissLabor")

# recode factors
sl <- SwissLabor
sl$participation <- as.numeric(sl$participation)-1
sl$foreign       <- as.numeric(sl$foreign)-1

X_new <- data.frame("income"=c(10,12), "age"=c(4.5,5.8))

## Not run:
fit1 <- TSVC(participation~income+age, data=sl, family=binomial(link="logit"),
             nperm=1000, trace=TRUE)
predict(fit1, X_new, type="response")

## End(Not run)
```

---

summary.TSVC *Summary of Tree-Structured Varying Coefficient Models*

Description

Summary for an object of class TSVC, with an overview of all executed splits during the fitting process.

Usage

```
## S3 method for class 'TSVC'
summary(object, ...)

## S3 method for class 'summary.TSVC'
print(x, ...)
```

Arguments

`object` object of class `TSVC`.

`...` further arguments passed to or from other methods.

`x` object of class `summary.TSVC`.

Value

object of class `"summary.TSVC"`. An object of class `"summary.TSVC"` is a list containing the following components:

`stats` overview of detected varying coefficients, responsible effect modifiers and executed splits.

`nosplits` total number of executed splits during the fitting process.

Author(s)

Moritz Berger <Moritz.Berger@imbie.uni-bonn.de>
`https://www.imbie.uni-bonn.de/personen/dr-moritz-berger/`

References

Berger, M., G. Tutz and M. Schmid (2019). Tree-Structured Modelling of Varying Coefficients. Statistics and Computing 29, 217-229, https://doi.org/10.1007/s11222-018-9804-8.

See Also

`TSVC`, `plot.TSVC`, `predict.TSVC`

Examples

```
# Swiss Labour Market
library(AER)
data("SwissLabor")

# recode factors
sl <- SwissLabor
sl$participation <- as.numeric(sl$participation)-1
sl$foreign       <- as.numeric(sl$foreign)-1

## Not run:
fit1 <- TSVC(participation~income+age, data=sl, family=binomial(link="logit"),
             nperm=1000, trace=TRUE)
summary(fit1)

## End(Not run)
```

TSVC *Tree-Structured Modelling of Varying Coefficients*

Description

A function to fit tree-structured varying coefficient (TSVC) models. By recursive splitting the method allows to simultaneously detect covariates with varying coefficients and the effect modifiers that induce varying coefficients if they are present. The basic method is described in Berger, Tutz and Schmid (2018).

Usage

```
TSVC(
  formula,
  data,
  family = gaussian,
  alpha = 0.05,
  nperm = 1000,
  nodesize_min = 5,
  bucket_min = 1,
  depth_max = NULL,
  perm_test = TRUE,
  effmod = NULL,
  notmod = NULL,
  only_effmod = NULL,
  smooth = NULL,
  split_intercept = FALSE,
  trace = FALSE,
  ...
)

## S3 method for class 'TSVC'
print(x, ...)
```

Arguments

`formula` object of class `formula`: a symbolic description of the (linear) model to be fit. See also details.

`data` data frame of class `data.frame` containing the variables in the model.

`family` a description of the error distribution and link function to be used in the model (as for `glm`). This can be a character string naming a family function, a family function or the result of a call to a family function. See `family` for details of family functions.

`alpha` significance level $alpha$ for the permutation tests.

`nperm` number of permutations used for the permutation tests.

`nodesize_min` minimum number of observations that must exist in a node in order for a split to be attempted.

`bucket_min` the minimum number of observations in any terminal node.

`depth_max` maximum depth of any node in each tree, with the root node counted as depth 0. If `NULL` (default), the size of the trees is not restricted.

`perm_test` if `FALSE`, no permutation tests are performed, but each tree is grown until the minimum node size constraint is reached.

`effmod` optional vector of covariates that serve as effect modifier. If `NULL` (default), all covariates are considered as potential effect modifiers.

`notmod` optional list of class `list` containing pairs of covariate/effect modifier that are not considered as candidates for splitting during iteration. If `NULL` (default), all combinations of covariates and potential effect modifiers are considered for splitting.

`only_effmod` optional vector of covariates that serve as effect modifier, only. If `NULL` (default), all effect modifiers are included in the predictor of the model and are allowed to be modified.

`smooth` optional vector of covariates with a smooth effect on the response. The (smooth) effects fo these variables are not allowed to be modified.

`split_intercept`
if `TRUE`, the intercept is allowed to be modified by the covariates. If `FALSE` (default), the intercept is set constant.

`trace` if `TRUE`, information about the estimation progress is printed.

`...` further arguments passed to or from other methods.

`x` object of class `TSVC`.

Details

A typical formula has the form `response ~ covariates`, where `response` is the name of the response variable and `covariates` is a series of variables that are incorporated in the model.

With p covariates, `TSVC` expects a formula of the form y $x_1 + ... + x_p$. If no further specifications are made (`effmod=NULL`, `notmod=NULL`, `only_effmod=NULL`) it is assumed that each covariate $x_j, j = 1, ..., p$ can be modified by all the other variables $x_m, m = 1, ..., p$ j.

Remark: Significance of each split is verified by permutation tests. The result of the permutation tests can strongly depend on the number of permutations `nperm`.

Note: The algorithm currently does not support splitting of/by factor variables. If a factor variable is included in the formula of the model, the variable will not serve as effect modifier and its effect will not be modified.

Value

Object of class `"TSVC"`. An object of class `"TSVC"` is a list containing the following components:

`splits` matrix with detailed information about all executed splits during the fitting process.

`coefficients` list of estimated coefficients for covariates with and without varying coefficients (including a non-varying intercept).

`pvalues` p-values of each permuation test during the fitting process.

`pvalues_linear` p-values of the permutation tests on the linear effects in the last step of the algorithm.

`devs` maximal value statistics T_m of the selected effect modifier in each iteration during the fitting process.

`crit` critical values of each permutation test during the fitting process.

`y` response vector.

`X` matrix of all the variables (covariates and effect modifiers) for model fitting.

`model` internally fitted model in the last iteration of class `glm` or `gam`.

Author(s)

Moritz Berger <Moritz.Berger@imbie.uni-bonn.de>
https://www.imbie.uni-bonn.de/personen/dr-moritz-berger/

References

Berger, M., G. Tutz and M. Schmid (2019). Tree-Structured Modelling of Varying Coefficients. Statistics and Computing 29, 217-229, https://doi.org/10.1007/s11222-018-9804-8.

Hastie, T. and R. Tibshirani (1993). Varying-coefficient models. Journal of the Royal Statistical Society B 55, 757-796.

Hothorn T., K. Hornik and A. Zeileis (2006). Unbiased recursive partitioning: A conditional inference framework. Journal of Computational and Graphical Statistics 15(3), 651-674.

See Also

`plot.TSVC`, `predict.TSVC`, `summary.TSVC`

Examples

```
# Swiss Labour Market
library(AER)
data("SwissLabor")

# recode factors
sl <- SwissLabor
sl$participation <- as.numeric(sl$participation)-1
sl$foreign       <- as.numeric(sl$foreign)-1

## Not run:
fit1 <- TSVC(participation~income+age, data=sl, family=binomial(link="logit"),
             nperm=300, trace=TRUE)
print(fit1)
class(fit$model) # glm

# In fit2, variable 'foreign' does not serve as effect modifier
# and the effect of 'foreign' is not modified by the other variables.
# That means 'foreign' is assumed to only have simple linear effect on the response.
fit2 <- TSVC(participation~income+age+foreign, data=sl, family=binomial(link="logit"),
             nperm=300, trace=TRUE, effmod=c("income","age"),
             notmod=list(c("foreign","income"),c("foreign","age")))
print(fit2)

# In fit3, variable 'age' does only serve as effect modifier. That means the effect of 'age'
# is not included in the predictor of the model.
fit3 <- TSVC(participation~income+age+foreign, data=sl, family=binomial(link="logit"),
             nperm=300, trace=TRUE, only_effmod="age")
print(fit3)

# In fit4, the intercept is allowed to be modified by 'age' and 'income'.
# The two covariates, however, are not allowed to modify each other.
fit4 <- TSVC(participation~income+age, data=sl, family=binomial(link="logit"),
             nperm=300, trace=TRUE, split_intercept=TRUE,
             notmod=list(c("income","age"), c("age", "income")))
print(fit4)


# In fit5, variable 'age' has a smooth effect on the response.
# Hence, the (smooth) effect of 'age' will not be modified by the other variables.
fit5 <- TSVC(participation~income+age+foreign, data=sl, family=binomial(link="logit"),
             nperm=300, trace=TRUE, smooth="age")
print(fit5)
class(fit5$model) # gam

# In fit6, the intercept is allowed to be modified by 'age' and 'income', but the two variables are
# not included in the predictor of the model. Here, no permutation tests are performed, but the
# tree is pruned by a minimum node size constraint.
fit6 <- TSVC(participation~income+age, data=sl, family=binomial(link="logit"),
         perm_test=FALSE, nodesize_min=100, bucket_min=100, trace=TRUE, split_intercept=TRUE,
             effmod=c("income","age"), only_effmod = c("income", "age"))
print(fit6)


## End(Not run)
```

Index