Package ‘psica’

October 14, 2022

Type Package

Title Decision Tree Analysis for Probabilistic Subgroup Identification with Multiple Treatments

Version 1.0.2

Author Oleg Sysoev, Krzysztof Bartoszek, Katarina Ekholm Selling and Lotta Ekstrom

Maintainer Oleg Sysoev <Oleg.Sysoev@liu.se>

Description In the situation when multiple alternative treatments or interventions available, different population groups may respond differently to different treatments. This package implements a method that discovers the population subgroups in which a certain treatment has a better effect than the other alternative treatments. This is done by first estimating the treatment effect for a given treatment and its uncertainty by computing random forests, and the resulting model is summarized by a decision tree in which the probabilities that the given treatment is best for a given subgroup is shown in the corresponding terminal node of the tree.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports Rdpack, grid, gridBase, randomForest, rpart, partykit, party, BayesTree

RdMacros Rdpack

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-02-11 17:20:11 UTC

R topics documented:

`plot.psicaTree` *Plots a PSICA tree.*

Description

A plot is produced that shows a PSICA tree in which the terminal nodes contain the probabilities that one treatment is better than the other ones as well as a label containing the possible best treatments

Usage

```
## S3 method for class 'psicaTree'
plot(x, type = 1, ...)
```

Arguments

`x`	the PSICA tree to be plotted.
`type`	If type=1 a default plot showing the probability of each treatment to be the best are displayed, if type=2 boxplots of the effect values are displayed in each terminal node.
`...`	Other parameter passed to the generic plot function

`predict.psicaTree` *Makes predictions for PSICA tree.*

Description

Makes predictions for PSICA tree.

Usage

```
## S3 method for class 'psicaTree'
predict(object, prob = FALSE, ...)
```

Arguments

`object`	the PSICA tree to be predicted
`prob`	should the matrix or class probabilities be predicted (TRUE) or the classes themselves (FALSE)
`...`	further parameters to be passed to ’rpart’ object.

print.psicaTree *Prints a PSICA tree.*

Description

Prints a PSICA tree.

Usage

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

Arguments

x	the PSICA tree to be printed
...	Other parameter passed to the generic print function

prune.psicaTree *Prunes a PSICA tree.*

Description

Prunes a PSICA tree.

Usage

```
## S3 method for class 'psicaTree'
prune(pT, cp = 0.001, ...)
```

Arguments

pT	the PSICA tree to be pruned.
cp	Cost complexity parameter that defines how much the tree must be pruned. See rpart::control() for description of the cost-complexity parameter.
...	further parameters to be passed to 'rpart' object.

psica — *Create a tree that discovers groups having similar treatment (intervention) effects.*

Description

The PSICA method operates by first building regression trees for each treament group and then obtaining the distributions of the effect size for given levels of independent variables by either bootstrap or by means of the bias-corrected infinitesimal jackknife. The obtained distributions are used for computing the probabilities that one treatment is better (effect size is greater) than the other treatments for a given set of input values. These probabilities are then summarised in the form of a decision tree built with a special loss function. The terminal nodes of the resulting tree show the probabilities that one treatment is better than the other treatments as well as a label containing the possible best treatments.

Usage

```
psica(formula, data, intervention, method = "normal",
  forestControl = list(minsplit = 10, mincriterion = 0.95, nBoots = 500,
  nTrees = 200, mtry = 5), treeControl = rpart::rpart.control(minsplit =
  20, minbucket = 10, cp = 0.003), confidence = 0.95, prune = TRUE,
  ...)
```

Arguments

`formula` Formula that shows the dependent variable (effect) and independent variables (separated by '+'). The treatment variable should not be present among dependent variables

`data` Data frame containing dependent and independent variables and the categorical treatment variable

`intervention` The name of the treatment variable

`method` Choose "boot" for computing probabilities by bootstrapping random forests, "normal" for computing probabilities by appoximating random forest variance with infinitesimal jackknife with bias correction.

`forestControl` parameters of forest growing, a list with parameters

- minsplit: minimum number of observation in the node to be splitted when growing random forest, default 10
- mincriterion: "mincriterion" setting of the random forest, see ctree in package partykit
- nBoots: number of trees in random forest.
- nTrees: amount of trees in each random forest
- mtry: number of variables to be selected at each split. Choose either 'sqrt(amount_of_inputs)' if amount of input variables is large or 'amount_of_inputs' if there are few input variables.

`treeControl` Parameters for decision tree growing, see rpart.control()

`confidence` Parameter that defines the cut-off probability in the loss function and also which treatments are included in the labels of the PSICA tree. More specifically, labels in the terminal nodes show all treatments except of useless treatments, i.e. the treatments that altogether have a probability to be the best which is smaller than 1-confidence.

`prune` should the final tree be pruned or is (possibly) overfitted tree desired?

`...` further argumets passed to rpart object.

Value

Object of a class psicaTree

References

Sysoev O, Bartoszek K, EkstrÃƒÂ¶m E, Ekholm Selling K (2019). “PSICA: Decision trees for probabilistic subgroup identification with categorical treatments.” *Statistics in Medicine*, **38**(22), 4436-4452. doi: 10.1002/sim.8308, https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.8308, `https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.8308`.

Examples

```
n=100
X1=runif(n)
X1=sort(X1)
f1<- function(x){
  2*tanh(4*x-2)+3
}
X2=runif(n)
X2=sort(X2)
f2<- function(x){
  2*tanh(2*x-1)+2.3 #2.8
}
plot(X1,f1(X1),ylim=c(0,5), type="l")
points(X2,f2(X2), type="l")
Y1=f1(X1)+rnorm(n, 0, 0.8)
Y2=f2(X2)+rnorm(n,0,0.8)
points(X1,Y1, col="blue")
points(X2,Y2, col="red")
data=data.frame(X=c(X1,X2), Y=c(Y1,Y2), interv=c(rep("treat",n), rep("control",n)))
pt=psica(Y~X, data=data, method="normal",intervention = "interv",
 forestControl=list(nBoots=200, mtry=1))
print(pt)
plot(pt)
```

snip.psicaTree *Prunes desired nodes in a PSICA tree.*

Description

Prunes desired nodes in a PSICA tree.

Usage

```
## S3 method for class 'psicaTree'
snip(pT, indices)
```

Arguments

pT	the PSICA tree to be pruned.
indices	vector of indices of the nodes. All contents below the nodes with these indices will be pruned.

Index