Package ‘pcatsAPIclientR’

October 14, 2022

Type Package

Title 'PCATS' API Client

Version 1.0.0

Author Bin Huang [aut],
Chen Chen [aut],
Michal Kouril [aut, cre]

Maintainer Michal Kouril <`michal.kouril@cchmc.org`>

Description Provides an R interface to the 'PCATS' API
<`https://pcats.research.cchmc.org/api/__docs__/`>,
allowing R users to submit tasks and retrieve results.

License GNU General Public License

Encoding UTF-8

RoxygenNote 7.1.1

Imports httr, jsonlite, methods, testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2021-12-17 10:20:02 UTC

R topics documented:

`dynamicGP` *Performs a data analysis for data with adaptive treatments.*

Description

Performs Bayesian's Gaussian process regression or Bayesian additive regression tree for data with adaptive treatment(s).

Usage

```
dynamicGP(
  datafile = NULL,
  dataref = NULL,
  method = "BART",
  stg1.outcome,
  stg1.treatment,
  stg1.x.explanatory = NULL,
  stg1.x.confounding = NULL,
  stg1.tr.hte = NULL,
  stg1.tr.values = NULL,
  stg1.tr.type = "Discrete",
  stg1.outcome.type = "Continuous",
  stg1.outcome.bound_censor = "neither",
  stg1.outcome.lb = NULL,
  stg1.outcome.ub = NULL,
  stg1.outcome.censor.lv = NULL,
  stg1.outcome.censor.uv = NULL,
  stg1.outcome.censor.yn = NULL,
  stg1.outcome.link = "identity",
  stg1.pr.values = NULL,
  stg2.outcome,
  stg2.treatment,
  stg2.x.explanatory = NULL,
  stg2.x.confounding = NULL,
  stg2.tr1.hte = NULL,
  stg2.tr2.hte = NULL,
  stg2.tr.values = NULL,
  stg2.tr.type = "Discrete",
  stg2.outcome.type = "Continuous",
  stg2.outcome.bound_censor = "neither",
  stg2.outcome.lb = NULL,
  stg2.outcome.ub = NULL,
  stg2.outcome.censor.lv = NULL,
  stg2.outcome.censor.uv = NULL,
  stg2.outcome.censor.yn = NULL,
  stg2.outcome.link = "identity",
  stg2.pr.values = NULL,
```

```
  burn.num = 500,
  mcmc.num = 500,
  x.categorical = NULL,
  mi.datafile = NULL,
  mi.dataref = NULL,
  sheet = NULL,
  mi.sheet = NULL,
  seed = 5000,
  token = NULL,
  use.cache = NULL
)
```

Arguments

`datafile`	File to upload (.csv or .xls)
`dataref`	Reference to already uploaded file.
`method`	The method to be used. "GP" for GP method and "BART" for BART method. The default value is "BART".
`stg1.outcome`	The name of the intermediate outcome variable for stage 1.
`stg1.treatment`	The name of the treatment variable for stage 1.
`stg1.x.explanatory`	A vector of the name of the explanatory variables for stage 1.
`stg1.x.confounding`	A vector of the name of the confounding variables for stage 1.
`stg1.tr.hte`	An optional vector specifying categorical variables which may have heterogeneous treatment effect with the treatment variable for stage 1.
`stg1.tr.values`	User-defined values for the calculation of ATE if the treatment variable is continuous for stage 1.
`stg1.tr.type`	The type of treatment at stage 1. "Continuous" for continuous treatment and "Discrete" for categorical treatment. The default value is "Discrete".
`stg1.outcome.type`	Intermediate outcome type ("Continuous" or "Discrete") for stage 1.
`stg1.outcome.bound_censor`	The default value is "neither". "neither" if the intermediate outcome is not bounded or censored. "bounded" if the intermediate outcome is bounded. "censored" if the intermediate outcome is censored.
`stg1.outcome.lb`	Stage 1 lower bound if the intermediate outcome is bounded.
`stg1.outcome.ub`	Stage 1 upper bound if the intermediate outcome is bounded.
`stg1.outcome.censor.lv`	lower variable of censored interval if the intermediate outcome is censored.
`stg1.outcome.censor.uv`	upper variable of censored interval if the intermediate outcome is censored.
`stg1.outcome.censor.yn`	Censoring variable if the intermediate outcome is censored.

`stg1.outcome.link`	function for the intermediate outcome; the default value is “identity”. "identity" if no transformation needed. "log" for log transformation. "logit" for logit transformation.
`stg1.pr.values`	An optional vector of user-defined values of c for PrTE at stage 1.
`stg2.outcome`	The name of the outcome variable for stage 2.
`stg2.treatment`	The name of the treatment variable for stage 2.
`stg2.x.explanatory`	A vector of the name of the explanatory variables for stage 2.
`stg2.x.confounding`	A vector of the name of the confounding variables for stage 2.
`stg2.tr1.hte`	At stage 2, an optional vector specifying cate-gorical variables which may have heterogeneoustreatment effect with the stage 1 treatment variable
`stg2.tr2.hte`	At stage 2, an optional vector specifying cate-gorical variables which may have heterogeneoustreatment effect with the stage 2 treatment variable
`stg2.tr.values`	User-defined values for the calculation of ATE if the treatment variable is continuous for stage 2.
`stg2.tr.type`	The type of treatment at stage 2. "Continuous" for continuous treatment and "Discrete" for categorical treatment. The default value is "Discrete".
`stg2.outcome.type`	Outcome type ("Continuous" or "Discrete") for stage 2.
`stg2.outcome.bound_censor`	The default value is "neither". "neither" if the intermediate outcome is not bounded or censored. "bounded" if the intermediate outcome is bounded. "censored" if the intermediate outcome is censored.
`stg2.outcome.lb`	Stage 2 lower bound if the outcome is bounded.
`stg2.outcome.ub`	Stage 2 upper bound if the outcome is bounded.
`stg2.outcome.censor.lv`	lower variable of censored interval if the outcome is censored.
`stg2.outcome.censor.uv`	upper variable of censored interval if the outcome is censored.
`stg2.outcome.censor.yn`	Censoring variable if the outcome is censored.
`stg2.outcome.link`	function for the outcome; the default value is “identity”. "identity" if no transformation needed. "log" for log transformation. "logit" for logit transformation.
`stg2.pr.values`	An optional vector of user-defined values of c for PrTE at stage 2.
`burn.num`	numeric; the number of MCMC ’burn-in’ samples, i.e. number of MCMC to be discarded. The default value is 500.
`mcmc.num`	numeric; the number of MCMC samples after ’burn-in’. The default value is 500.
`x.categorical`	A vector of the name of categorical variables in data.

`mi.datafile`	File to upload (.csv or .xls) that contains the imputed data in the model.
`mi.dataref`	Reference to already uploaded file that contains the imputed data in the model.
`sheet`	If `datafile` or `dataref` points to an Excel file this variable specifies which sheet to load.
`mi.sheet`	If `mi.datafile` or `mi.dataurl` points to an Excel file this variable specifies which sheet to load.
`seed`	Sets the seed. The default value is 5000.
`token`	Authentication token.
`use.cache`	Use cached results (default True).

Value

jobid

`dynamicGP.cate` *Get conditional average treatment effect for data with two time points.*

Description

Estimate the conditional average treatment effect of user-specified treatment groups.

Usage

```
dynamicGP.cate(
  jobid,
  x,
  control.tr,
  treat.tr,
  pr.values = NULL,
  token = NULL,
  use.cache = NULL
)
```

Arguments

`jobid`	job id of the "dynamicGP".
`x`	The name of variable which may have the heterogeneous treatment effect. `x` should be a categorical variable.
`control.tr`	A vector of the values of the treatment variables at all stages as the reference group.
`treat.tr`	A vector of the values of the treatment variables at all stages compared to the reference group.
`pr.values`	An optional vector of user-defined values of c for PrCTE.
`token`	Authentication token.
`use.cache`	Use cached results (default True).

Details

The contrast of potential outcomes for the reference group and the treatment group is estimated at a list of x values if x is not a factor. If x is a factor, the conditional average treatment effect is estimated at each value of levels of x.

Value

jobid

Note

The conditional average treatment effect is estimated based on the sample data. The observations with missing covariates in the model are excluded. For the unspecified variables in the model, the observed data is used to estimate the conditional average treatment effect.

`job_status` *Return job status*

Description

Return status of the previously submitted job

Usage

```
job_status(jobid, token = NULL)
```

Arguments

`jobid`	Job ID of the previously submitted job
`token`	Authentication token.

Value

status

`ploturl` *Return plot URL*

Description

Return plot URL

Usage

```
ploturl(jobid, plottype = "", token = NULL)
```

Arguments

`jobid`	Job ID of the previously submitted job
`plottype`	Plot type
`token`	Authentication token.

Value

url

`printgp` *Print job results*

Description

Return formatted string with job results

Usage

```
printgp(jobid, token = NULL)
```

Arguments

`jobid`	Job ID of the previously submitted job
`token`	Authentication token.

Value

formatted text

`results` *Return job results*

Description

Return job results

Usage

```
results(jobid, token = NULL)
```

Arguments

`jobid`	Job ID of the previously submitted job
`token`	Authentication token.

Value

results

`staticGP` *Performs a data analysis for data with non-adaptive treatment(s).*

Description

Performs Bayesian's Gaussian process regression or Bayesian additive regression tree for data with non-adaptive treatment(s).

Usage

```
staticGP(
  datafile = NULL,
  dataref = NULL,
  method = "BART",
  outcome,
  outcome.type = "Continuous",
  outcome.bound_censor = "neither",
  outcome.lb = NULL,
  outcome.ub = NULL,
  outcome.censor.yn = NULL,
  outcome.censor.lv = NULL,
  outcome.censor.uv = NULL,
  outcome.link = "identity",
  treatment,
  x.explanatory = NULL,
```

```
  x.confounding = NULL,
  tr.type = "Discrete",
  tr2.type = "Discrete",
  tr.values = NULL,
  tr2.values = NULL,
  pr.values = NULL,
  tr.hte = NULL,
  tr2.hte = NULL,
  burn.num = 500,
  mcmc.num = 500,
  x.categorical = NULL,
  mi.datafile = NULL,
  mi.dataref = NULL,
  sheet = NULL,
  mi.sheet = NULL,
  seed = 5000,
  token = NULL,
  use.cache = NULL
)
```

Arguments

`datafile`	File to upload (.csv or .xls)
`dataref`	Reference to already uploaded file.
`method`	The method to be used. "GP" for GP method and "BART" for BART method. The default value is "BART".
`outcome`	The name of the outcome variable.
`outcome.type`	Outcome type ("Continuous" or "Discrete"). The default value is "Continuous".
`outcome.bound_censor`	The default value is "neither". "neither" if the outcome is not bounded or censored. "bounded" if the outcome is bounded. "censored" if the outcome is censored.
`outcome.lb`	Putting a lower bound if the outcome is bounded.
`outcome.ub`	Putting a upper bound if the outcome is bounded.
`outcome.censor.yn`	Censoring variable if outcome is censored.
`outcome.censor.lv`	lower variable of censored interval if outcome is censored.
`outcome.censor.uv`	upper variable of censored interval if outcome is censored.
`outcome.link`	function for outcome; the default value is "identity". "identity" if no transformation needed. "log" for log transformation. "logit" for logit transformation.
`treatment`	The vector of the name of the treatment variables. Users can input at most two treatment variables.
`x.explanatory`	The vector of the name of the explanatory variables.

`x.confounding`	The vector of the name of the confounding variables.
`tr.type`	The type of the first treatment. "Continuous" for continuous treatment and "Discrete" for categorical treatment. The default value is "Discrete".
`tr2.type`	The type of the second treatment if available. "Continuous" for continuous treatment and "Discrete" for categorical treatment. The default value is "Discrete".
`tr.values`	user-defined values for the calculation of ATE if the first treatment variable is continuous
`tr2.values`	user-defined values for the calculation of ATE if the second treatment variable is continuous
`pr.values`	An optional vector of user-defined values of c for PrTE.
`tr.hte`	An optional vector specifying variables which may have heterogeneous treatment effect with the first treatment variable
`tr2.hte`	An optional vector specifying variables which may have heterogeneous treatment effect with the second treatment variable
`burn.num`	numeric; the number of MCMC 'burn-in' samples, i.e. number of MCMC to be discarded. The default value is 500.
`mcmc.num`	numeric; the number of MCMC samples after 'burn-in'. The default value is 500.
`x.categorical`	A vector of the name of categorical variables in data.
`mi.datafile`	File to upload (.csv or .xls) that contains the imputed data in the model.
`mi.dataref`	Reference to already uploaded file that contains the imputed data in the model.
`sheet`	If `datafile` or `dataref` points to an Excel file this variable specifies which sheet to load.
`mi.sheet`	If `mi.datafile` or `mi.dataurl` points to an Excel file this variable specifies which sheet to load.
`seed`	Sets the seed. The default value is 5000.
`token`	Authentication token.
`use.cache`	Use cached results (default True).

Value

jobid

`staticGP.cate` *Get conditional average treatment effect*

Description

Estimate the conditional average treatment effect of user-specified treatment groups.

Usage

```
staticGP.cate(
  jobid,
  x,
  control.tr,
  treat.tr,
  pr.values = NULL,
  token = NULL,
  use.cache = NULL
)
```

Arguments

`jobid`	job id of the "staticGP".
`x`	The name of a categorical variable which may have the heterogeneous treatment effect.
`control.tr`	The value of the treatment variable as the reference group.
`treat.tr`	The value of the treatment variable compared to the reference group.
`pr.values`	An optional vector of user-defined values of c for PrCTE.
`token`	Authentication token.
`use.cache`	Use cached results (default True).

Details

The contrast of potential outcomes for the reference group and the treatment group is estimated at each value of x.

Value

Return jobid

Note

The conditional average treatment effect is estimated based on the sample data. The observations with missing covariates in the model are excluded. For the unspecified variables in the model, the original data is used to estimate the conditional average treatment effect.

`uploadfile` *Upload a file*

Description

Upload a file

Usage

```
uploadfile(filename, token = NULL)
```

Arguments

`filename`	Filename of a file to upload
`token`	Authentication token.

Value

Backend filename reference

`wait_for_result` *Wait while the job status is pending*

Description

Return when the job status is finished (either successfully or otherwise)

Usage

```
wait_for_result(jobid, token = NULL)
```

Arguments

`jobid`	Job ID of the previously submitted job
`token`	Authentication token.

Value

status

Index