Package ‘pda’

October 14, 2022

Type Package

Title Privacy-Preserving Distributed Algorithms

Version 1.0-2

Date 2020-12-10

Description A collection of privacy-preserving distributed algorithms for conducting multi-site data analyses. The regression analyses can be linear regression for continuous outcome, logistic regression for binary outcome, Cox proportional hazard regression for time-to event outcome, or Poisson regression for count outcome. The PDA algorithm runs on a lead site and only requires summary statistics from collaborating sites, with one or few iterations. For more information, please visit our software websites: <https://github.com/Penncil/pda>, and <https://pdamethods.org/>.

Maintainer Chongliang Luo <luocl3009@gmail.com>

License Apache License 2.0

Suggests imager

Imports Rcpp (>= 0.12.19), stats, htr, rvest, jsonlite, data.table, survival

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation yes

Author Chongliang Luo [aut, cre],
Rui Duan [aut],
Mackenzie Edmondson [aut],
Jiayi Tong [aut],
Yong Chen [aut],
Penn Computing Inference Learning (PennCIL) lab [cph]

Repository CRAN

Date/Publication 2020-12-10 21:20:02 UTC
Description

gather cloud settings into a list

Usage

getCloudConfig(site_id, dir, uri, secret)

Arguments

site_id site identifier
dir shared directory path if flat files
uri web uri if web service
secret web token if web service

Value

A list of cloud parameters: site_id, secret and uri

See Also

pda
Description

Fit Privacy-preserving Distributed Algorithms for linear, logistic, Poisson and Cox PH regression with possible heterogeneous data across sites.

Usage

pda(ipdata, site_id, control, dir, uri, secret)

Arguments

ipdata  Local IPD data in data frame, should include at least one column for the outcome and one column for the covariates
site_id  Character site name
control  pda control data
dir  directory for shared flat file cloud
uri  Universal Resource Identifier for this run
secret  password to authenticate as site_id on uri

Value

control
control

References


See Also

pdaPut, pdaList, pdaGet, getCloudConfig and pdaSync.
Examples

```
require(survival)
require(data.table)
require(pda)
data(lung)

## In the toy example below we aim to analyze the association of lung status with
## age and sex using logistic regression, data(lung) from 'survival', we randomly
## assign to 3 sites: 'site1', 'site2', 'site3'. we demonstrate using PDA ODAL can
## obtain a surrogate estimator that is close to the pooled estimate. We run the
## example in local directory. In actual collaboration, account/password for pda server
## will be assigned to the sites at the server https://pda.one.
## Each site can access via web browser to check the communication of the summary stats.

## for more examples, see demo(ODAC) and demo(ODAP)

# Create 3 sites, split the lung data amongst them
sites = c('site1', 'site2', 'site3')
set.seed(42)
lung2 <- lung[,c('status', 'age', 'sex')]
lung2$sex <- lung2$sex - 1
lung2$status <- ifelse(lung2$status == 2, 1, 0)
lung_split <- split(lung2, sample(1:length(sites), nrow(lung), replace=TRUE))

## fit logistic reg using pooled data
fit.pool <- glm(status ~ age + sex, family = 'binomial', data = lung2)

# ###################################################################
# STEP 1: initialize
# ###################################################################
control <- list(project_name = 'Lung cancer study',
                step = 'initialize',
                sites = sites,
                heterogeneity = FALSE,
                model = 'ODAL',
                family = 'binomial',
                outcome = 'status',
                variables = c('age', 'sex'),
                optim_maxit = 100,
                lead_site = 'site1',
                upload_date = as.character(Sys.time()) )

# run the example in local directory:
# specify your working directory, default is the tempdir
mydir <- tempdir()
# assume lead site1: enter "1" to allow transferring the control file
pda(site_id = 'site1', control = control, dir = mydir)
# in actual collaboration, account/password for pda server will be assigned, thus:
# Not run: pda(site_id = 'site1', control = control, uri = 'https://pda.one', secret='abc123')
# you can also set your environment variables, and no need to specify them in pda:
# Not run: Sys.setenv(PDA_USER = 'site1', PDA_SECRET = 'abc123', PDA_URI = 'https://pda.one')
# Not run: pda(site_id = 'site1', control = control)
```
pdaGet

Function to download json and return as object

Description

Function to download json and return as object
pdaList

Usage

pdaGet(name, config)

Arguments

name of file
config cloud configuration

Value

A list of data objects from the json file on the cloud

See Also

pda

pdaList

Function to list available objects

Description

Function to list available objects

Usage

pdaList(config)

Arguments

config a list of variables for cloud configuration

Value

A list of (json) files on the cloud

See Also

pda
pdaPut

Function to upload object to cloud as json

Description

Function to upload object to cloud as json

Usage

pdaPut(obj, name, config)

Arguments

- obj: R object to encode as json and uploaded to cloud
- name: name of file
- config: a list of variables for cloud configuration

Value

NONE

See Also

pda

pdaSync

pda control synchronize

Description

update pda control if ready (run by lead)

Usage

pdaSync(config)

Arguments

- config: cloud configuration

Value

control

See Also

pda
Index

getCloudConfig, 2

pda, 3
pdaGet, 5
pdaList, 6
pdaPut, 7
pdaSync, 7