

Package: privreg (via r-universe)

November 24, 2024

Type Package

Title Privacy-preserving Regression

Version 0.9.5

Author Erik-Jan van Kesteren

Maintainer Erik-Jan van Kesteren <e.vankesteren1@uu.nl>

Description Generalized linear modeling on vertically partitioned data using block coordinate descent.

License GPL-3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Imports httpuv, websocket, R6, openssl, Matrix, stats, RSpectra, MASS

Remotes vankesteren/firatheme

RoxygenNote 6.1.1

Suggests testthat, firatheme, dplyr, ggplot2, tibble, tidyr

Config/pak/sysreqs make libssl-dev zlib1g-dev

Repository <https://vankesteren.r-universe.dev>

RemoteUrl <https://github.com/vankesteren/privreg>

RemoteRef HEAD

RemoteSha a05a014c67f6da755800f30678034e968a7c1981

Contents

PrivReg	2
privreg_local	4

Index	5
--------------	----------

PrivReg

Private regression with vertically partitioned data

Description

Perform privacy-preserving regression modeling across different institutions. This class implements regression with gaussian and binomial responses using block coordinate descent.

Value

an R6 object of class PrivReg

Usage

```
alice <- PrivReg$new(  
  formula,  
  data,  
  family   = "gaussian",  
  name     = "alice",  
  verbose  = FALSE,  
  debug    = FALSE,  
  crypt_key = "testkey"  
)
```

```
alice$listen()  
alice$connect(127.0.0.1)  
alice$disconnect()
```

```
alice$estimate()  
alice$calculate_se()
```

```
alice$summary()  
alice$coef()  
alice$converged()  
alice$plot_paths()  
alice$elapsed()
```

Arguments

- formula model formula for the regression model at this institution
- data data frame for the variables in the model formula
- family response family as in glm. Currently only gaussian and binomial are supported
- intercept whether to include the intercept. Always use this instead of + 0 in the model formula
- name name of this institution
- verbose whether to print information

- debug whether to print debug statements
- crypt_key pre-shared key used to encrypt communication

Details

- \$new() instantiates and returns a new PrivReg object.
- \$listen() listens for incoming connections from a partner institution
- \$connect() connects to a listening partner institution
- \$disconnect() disconnects from the partner institution
- \$set_control() sets control parameters. See below for more info
- \$estimate() computes parameter estimates through block coordinate descent
- \$calculate_se() computes standard errors using projection method
- \$converged() test whether the algorithm has converged
- \$summary() displays a summary of the object, invisibly returns the coef matrix
- \$coef() returns the model coefficients
- \$plot_paths() plots the paths of the parameters over the estimation iterations
- \$elapsed() print information about the elapsed time

Control parameters

- max_iter maximum number of iterations of the coordinate descent algorithm
- tol PrivReg is converged if all beta changes are below tol.
- se Whether to compute standard errors

Examples

```
## Not run:
# generate some data
set.seed(45)
X <- matrix(rnorm(1000), 100)
b <- runif(10, -1, 1)
y <- X %*% b + rnorm(100, sd = sqrt(b %*% S %*% b))

# split into alice and bob institutions
alice_data <- data.frame(y, X[, 1:5])
bob_data <- data.frame(y, X[, 6:10])

# create connection
alice$listen()
bob$connect("127.0.0.1") # if alice is on different computer, change ip

# estimate
alice$estimate()

# ...

# compare results to lm()
```

```
summary(lm(y ~ X + 0))
alice$summary()
bob$summary()

## End(Not run)
```

privreg_local *Local vertically partitioned data regression*

Description

Perform privreg locally with two vertically partitioned datasets

Usage

```
privreg_local(y, Xa, Xb, family = gaussian(), se = TRUE, tol = 1e-12,
             maxit = 10000, debug = TRUE)
```

Arguments

y	outcome variable
Xa	alice model matrix
Xb	bob model matrix
family	response family (use family object!)
se	whether to compute the standard error
tol	tolerance
maxit	maximum iterations
debug	print debug information

Index

PrivReg, [2](#)
privreg_local, [4](#)