Package 'pql'

May 21, 2024

Type Package

Title A Partitioned Quasi-Likelihood for Distributed Statistical Inference

Version 0.1.0

Author Guangbao Guo [aut, cre], Jiarui Li [aut]

Maintainer Guangbao Guo <ggb11111111@163.com>

Description In the big data setting, working data sets are often distributed on multiple machines. However, classical statistical methods are often developed to solve the problems of single estimation or inference. We employ a novel parallel quasi-likelihood method in generalized linear models, to make the variances between different sub-estimators relatively similar. Estimates are obtained from projection subsets of data and later combined by suitably-chosen unknown weights. The philosophy of the package is described in Guo G. (2020) <doi:10.1007/s00180-020-00974-4>.

License MIT + file LICENSE

Encoding UTF-8

Imports parallel,pracma

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2024-05-21 14:20:02 UTC

R topics documented:

pqlBLogist *The weighted Gauss-Newton estimators of the PQL in Logistic-GLMs*

Description

The average weighted estimator and the unknown weighted estimator of the PQL in Logistic-GLMs through damped Gauss-Newton updates.

Usage

```
pqlBLogist(data,G,nk)
```

Arguments

data	is a design matrix with uniform distribution and the response vector.
G	is the number of subsets.
nk	is the size of subsets.

Value

betaBW,betaBA,MSEW,MSEA

Examples

```
G <- 20;n=1000;p=5; nk=50
b=runif(p, 0, 1)
beta =matrix(b,nrow=p)
X=matrix(rnorm(n*p),nrow=n)
L=X%*%beta
prob=1/exp(-(0.48+(L))+1)
y=1/(1+exp(-X))
y=(prob>runif(n))
y= ifelse((prob>runif(n)), 1, 0)
data=cbind(y,X)
pqlBLogist(data,G,nk)
```

pqlBpoisson1 *The weight Gauss-Newton estimators of the PQL in Poisson-GLMS*

Description

The average weighted estimator and the unknown weighted estimator of the PQL in Poisson-GLMS through damped Gauss-Newton

Usage

```
pqlBpoisson1(data,G,nk)
```

Arguments

data	is a design matrix with uniform distribution and the response vector
G	is the number of subsets.
nk	is the size of subsets

Value

betaBA, betaBW, MSEA, MSEW

Examples

```
G <- 20;n=1000;p=5; nk=50
X<- matrix(runif(1000* 5, 0, 0.5), ncol = 5)
beta =matrix(runif(p, 0, 1),nrow=p)
L=X%*%beta
y<- rpois(1000, exp(L))
data=cbind(y,X)
pqlBpoisson1(data,G,nk)
```

---

pqlBpoisson2 *The weighted Gauss-Newton estimators of the PQL in Poisson-GLMS*

Description

The average weighted estimator and the unknown weighted estimator of the PQL in Poisson-GLMS through damped Gauss-Newton

Usage

```
pqlBpoisson2(data,G,nk)
```

Arguments

data	is a design matrix with uniform distribution and the response vector
G	is the number of subsets.
nk	is the size of subsets.

Value

betaBA, betaBW, MSEA, MSEW

Examples

```
p<- 5;G<- 20;n<- 1000;nk=50
X<- matrix(runif(n * p, 0, 0.5), ncol = p)
beta =matrix(runif(p, 0, 1),nrow=p)
L=X%*%beta
y<- rpois(n, exp(L))
data=cbind(y,X)
pqlBpoisson2(data,G,nk)
```

---

pqlLogist *pqlLogist*

Description

The average weighted estimator and the unknown weighted estimator of the PQL in Poisson-GLMS through damped Gauss-Newton

Usage

```
pqlLogist(data,G,nk)
```

Arguments

data	data is a highly correlated data set
G	G is the number of nodes
nk	n1 is the length of each data subset

Value

betaW	estimation value of betaW
betaA	estimation value of betaA
MSEW	estimation of MSEW
MSEA	estimation of MSEA

Examples

```
p<- 5;G<- 20;n<- 1000;nk=200
X<- matrix(runif(n*p, 0, 0.5), ncol = p)
beta =matrix(runif(p, 0, 1),nrow=p)
L=X%*%beta
y<- rpois(n, exp(L))
data=cbind(y,X)
pqlLogist(data,G,nk)
```

pqlPoisson *The weighted Gauss-Newton estimators of the PQL in Poisson-GLMs*

Description

The average weighted estimator and the unknown weighted estimator of the PQL in Poisson-GLMS through damped Gauss-Newton

Usage

```
pqlPoisson(data,G,nk)
```

Arguments

data	is a design matrix with uniform distribution and the response vector
G	is the number of subsets
nk	is the number of outer subsets.

Value

betaBA, betaBW, MSEA, MSEW

Examples

```
#library(parallel)
#library(numDeriv)
#library(Rmpi)
#install.packages("pracma");
#library(pracma)
p<- 5;G<- 20;n<- 1000;nk=200
X<- matrix(runif(n*p, 0, 0.5), ncol = p)
beta =matrix(runif(p, 0, 1),nrow=p)
L=X%*%beta
y<- rpois(n, exp(L))
data=cbind(y,X)
pqlPoisson(data,G,nk)
```

Index