Package ISLET

March 24, 2020

Type Package

Title ISLET

Version 0.1.0

Authors Anru Zhang, Yuetian Luo, Garvesh Raskutti, Ming Yuan

Maintainer Yuetian Luo <ylu086@wisc.edu>

Description This package can do fast low-rank tensor regression via importance sketching and fast low rank sparse tensor regression via importance sketching. The algorithm is quite scalable and can handle dimension ten million. See paper Zhang et al. (2020) for reference.

License What license is it under?

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports Rdpack

RdMacros Rdpack

R topics documented:

`ISLET` *ISLET for regular tensor regression, 3-d case*

Description

This function do ISLET under the regular low-tucker rank tensor regression case. For detail about the procedure, see Zhang et al 2019.

Usage

```
ISLET(X, y, r1, r2, r3, p1, p2, p3)
```

Arguments

`X`	input tensor, n*p1*p2*p3
`y`	input response y
`r`	input rank like r1,r2,r3
`p`	input dimension like p1, p2, p3

Value

Estimate of coefficient tensor A.

References

Zhang A, Luo Y, Raskutti G, Yuan M (2020). "ISLET Fast and Optimal Low-rank Tensor Regression via Importance Sketching." *SIAM Journal on Mathematics of Data Science*.

Examples

```
set.seed(2018)
n = c(1000); p = c(10); r = c(3); sig = c(5);
p1 = p2 = p3 = p
r1 = r2 = r3 = r
# Generate data
S.array = array(rnorm(r1*r2*r3), dim=c(r1,r2,r3)) # Core tensor
S = as.tensor(S.array)
E1 = matrix(rnorm(p1*r1), nrow = p1, ncol=r1)
E2 = matrix(rnorm(p2*r2), nrow = p2, ncol=r2)
E3 = matrix(rnorm(p3*r3), nrow = p3, ncol=r3)
X = as.tensor(array(rnorm(n*p1*p2*p3), dim=c(n,p1,p2,p3)))
# Parameter tensor
A = ttm(ttm(ttm(S, E1, 1), E2, 2), E3, 3)
eps = sig * rnorm(n) # Error
y = regression.product(X, A) + eps # response
snr <- sd(y)/sd(eps)
# Estimator
A_sketch <- sketch(X, y, r1, r2, r3, p1, p2, p3)
error <- fnorm(A_sketch$hatA - A)/ fnorm(A)
```

`matrix_ISLET` *Matrix ISLET, 2-d case*

Description

Matrix ISLET, 2-d case

Usage

```
matrix_ISLET(X, y, r, p1, p2)
```

Arguments

`X`	input tensor data, n*p1*p2
`y`	input response y
`r`	input rank like r
`p`	input dimension like p1, p2

Value

Estimate of coefficient tensor A.

Examples

```
# EXAMPLE ISLET in Matrix case
n = c(5000); p = c(50); r = c(3); sig = c(5);
# data generation
S = diag(abs(rnorm(r))) # Core matrix
E1 = matrix(rnorm(p1*r), nrow = p1, ncol=r)
E2 = matrix(rnorm(p2*r), nrow = p2, ncol=r)
X = as.tensor(array(rnorm(n*p1*p2), dim=c(n,p1,p2)))
A = E1 %*% S %*% t(E2)
eps = sig * rnorm(n) # Error
y = matrix.regression.product(X, A) + eps # response
# Estimation
A_sketch <- matrix_sketch(X, y, r, p1, p2)
hatA <- A_sketch[[1]]
norm(hatA - A, type = "F")/norm(A, type = "F") # error
```

`select_r` *Estimate Rank when rank is unknown- 3-d case*

Description

This function is used to estimate the rank r when the Tucker rank of tensor is unknown. For detail about the procedure, see reference Zhang et al 2019 and Zhang 2018 CROSS: Efficient low rank tensor completion.s

Usage

```
select_r(X, y, thres, r1, r2, r3, p1, p2, p3)
```

Arguments

`X`	input tensor, n*p1*p2*p3
`y`	input response y
`thres`	thresholding in the autoselection of rank r, suggestion 1-2.
`r`	input a conservative rank estimation r1_init,r2_init,r3_inits
`p`	input dimension like p1, p2, p3

Value

Estimate Tucker rank of the parameter tensor

References

Adummy A (2020). “Not avalable.” Failed to insert reference with key = zhang2020cross from package = ’ISLET’. Possible cause — missing or misspelled key.

Examples

```
# ISLET in the unknown rank setting
n = c(1000); p = c(10); r = c(3); sig = c(5);
p1 = p2 = p3 = p
r1 = r2 = r3 = r
thres = 1.5; # thresholding is greater than 1, and not too big. Suggestion is 1-2.
# data generate
A = ttm(ttm(ttm(S, E1, 1), E2, 2), E3, 3)
eps = sig * rnorm(n) # Error
y = regression.product(X, A) + eps # response
snr <- sd(y)/sd(eps)
# Get a estimate of rank
r1_init = r2_init = r3_init = 8
r_select <- select_r(X, y, thres, r1_init, r2_init, r3_init, p1, p2, p3)
# Do estimation
A_sketch <- sketch(X, y, r_select[1], r_select[2], r_select[3], p1, p2, p3)
#error
fnorm(A_sketch - A)/fnorm(A)
```

`sparse_ISLET` *Sparse ISLET*

Description

This is the main function to do ISLET in the sparse tensor regression case with known rank r, the output is the estimation error and the running time

Usage

```
sparse_ISLET(X, y, sparse_mode, r1, r2, r3, p1, p2, p3)
```

Arguments

`X`	input tensor, n*p1*p2*p3
`y`	input response y
`sparse_mode`	Indicator of the sparse mode of the coefficient tensor, i.e. c(TRUE, TRUE, TRUE)
`r`	input rank like r1,r2,r3
`p`	input dimension like p1, p2, p3

Value

Estimate of coefficient tensor A.

Examples

```
set.seed(2018)
library(gglasso)
# EXAMPLE: Sparse tensor regression
n = c(3000); p = c(20); r = c(3); sig = c(5); s = c(12)
p1 = p2 = p3 = p
r1 = r2 = r3 = r
s1 = s2 = s3 = s
# Generate data
S.array = array(rnorm(r1*r2*r3), dim=c(r1,r2,r3)) # Core tensor
S = as.tensor(S.array)
E1 = matrix(rnorm(p1*r1), nrow = p1, ncol=r1)
E2 = matrix(rnorm(p2*r2), nrow = p2, ncol=r2)
E3 = matrix(rnorm(p3*r3), nrow = p3, ncol=r3)
# Add sparsity structure
E1[(s1+1):p1,] <- 0
E2[(s2+1):p2,] <- 0
E3[(s3+1):p3,] <- 0
sparse_mode = c(TRUE, TRUE, TRUE);
X = as.tensor(array(rnorm(n*p1*p2*p3), dim=c(n,p1,p2,p3)))
A = ttm(ttm(ttm(S, E1, 1), E2, 2), E3, 3) # Parameter tensor
eps = sig * rnorm(n) # Error
y = regression.product(X, A) + eps # response
snr <- sd(y)/sd(eps)
sparse_sketch_result <- sparse_sketch(X, y, sparse_mode, r1, r2, r3, p1, p2, p3)
fnorm(sparse_sketch_result$hatA - A)/fnorm(A) # errorss
sparse_sketch_result$running_time
```

`sparse_select_r` *Estimate rank in the sparse ISLET case with unknown rank*

Description

This function is used to select rank r for the sparse tensor regression case when the rank is unknonw

Usage

```
sparse_select_r(X, y, r1, r2, r3, p1, p2, p3, thres, sparse_mode)
```

Arguments

`y`	input response y
`thres`	a thresholding in the autoselection of rank r. Suggestion value 1-2.
`r`	input conservative esimate rank
`p`	input dimension

Value

Estimated rank

References

Zhang A, Luo Y, Raskutti G, Yuan M (2020). "ISLET Fast and Optimal Low-rank Tensor Regression via Importance Sketching." *SIAM Journal on Mathematics of Data Science*.

Examples

```
set.seed(2018)
n = c(3000); p = c(20); r = c(3); sig = c(5); s = c(12);
thres = c(2) # thresholding is greater than 1, and not too big. Suggestion is 1-2.
p1 = p2 = p3 = p
r1 = r2 = r3 = r
s1 = s2 = s3 = s
# Generate data
S.array = array(rnorm(r1*r2*r3), dim=c(r1,r2,r3)) # Core tensor
S = as.tensor(S.array)
E1 = matrix(rnorm(p1*r1), nrow = p1, ncol=r1)
E2 = matrix(rnorm(p2*r2), nrow = p2, ncol=r2)
E3 = matrix(rnorm(p3*r3), nrow = p3, ncol=r3)
# Add sparsity structure
E1[(s1+1):p1,] <- 0
E2[(s2+1):p2,] <- 0
E3[(s3+1):p3,] <- 0
sparse_mode = c(TRUE, TRUE, TRUE);
X = as.tensor(array(rnorm(n*p1*p2*p3), dim=c(n,p1,p2,p3)))
A = ttm(ttm(ttm(S, E1, 1), E2, 2), E3, 3)
eps = sig * rnorm(n) # Error
y = regression.product(X, A) + eps # response
snr <- sd(y)/sd(eps)
r1_init = 10; r2_init = 10; r3_init = 10; # an initial conservative estimator of rank
r_select = sparse_select_r(X, y,r1, r2, r3, p1, p2, p3, thres, sparse_mode)
sparse_sketch_result <- sparse_sketch(X, y, sparse_mode, r_select[1], r_select[2], r_select[3], p1, p2, p3)
fnorm(sparse_sketch_result$hatA - A)/fnorm(A) # errorss
sparse_sketch_result$running_time
```

Index