Package ‘ttTensor’

March 6, 2019

Type Package

Title Tensor-Train Decomposition

Version 0.99.2

Date 2019-02-17

Author Koki Tsuyuzaki, Manabu Ishii, Itoshi Nikaido

Maintainer Koki Tsuyuzaki <k.t.the-answer@hotmail.co.jp>

Suggests testthat

Depends R (>= 3.5.0)

Imports methods, rTensor, tensorr, PTAk, Matrix

Description Tensor-train is a compact representation for higher-order tensors. Some algorithms for performing tensor-train decomposition are available such as TT-SVD, TT-WOPT, and TT-Cross. For the details of the algorithms, see I. V. Oseledets (2011) <doi:10.1137/090752286>, Yuan Longao, et al (2017) <arXiv:1709.02641>, I. V. Oseledets (2010) <doi:10.1016/j.laa.2009.07.024>.

License Artistic-2.0

URL https://github.com/rikenbit/ttTensor

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-06 12:10:04 UTC

R topics documented:

ttTensor-package 2
maxvol 3
skeleton.decomp 4
TTCross 5
TTSVD 6
TTWOPT 7

Index **9**

ttTensor-package *Tensor-Train Decomposition*

Description

Tensor-train is a compact representation for higher-order tensors. Some algorithms for performing tensor-train decomposition are available such as TT-SVD, TT-WOPT, and TT-Cross. For the details of the algorithms, see I. V. Oseledets (2011) <doi:10.1137/090752286>, Yuan Longao, et al (2017) <arXiv:1709.02641>, I. V. Oseledets (2010) <doi:10.1016/j.laa.2009.07.024>.

Details

The DESCRIPTION file:

Package:	ttTensor
Type:	Package
Title:	Tensor-Train Decomposition
Version:	0.99.2
Date:	2019-02-17
Author:	Koki Tsuyuzaki, Manabu Ishii, Itoshi Nikaido
Maintainer:	Koki Tsuyuzaki <k.t.the-answer@hotmail.co.jp>
Suggests:	testthat
Depends:	R (>= 3.5.0)
Imports:	methods, rTensor, tensorr, PTAk, Matrix
Description:	Tensor-train is a compact representation for higher-order tensors. Some algorithms for performing tensor-train
License:	Artistic-2.0
URL:	https://github.com/rikenbit/ttTensor

Index of help topics:

```
TTCross                 Tensor-Train Decomposition by TRCross
TTSVD                   Tensor-Train Decomposition by TTSVD
TTWOPT                  Tensor-Train Decomposition by Tensor-train
                        Weighted OPTimization
maxvol                  maxvol algorithm
skeleton.decomp         Skeleton Decomposition
ttTensor-package        Tensor-Train Decomposition
```

Author(s)

Koki Tsuyuzaki, Manabu Ishii, Itoshi Nikaido

Maintainer: Koki Tsuyuzaki <k.t.the-answer@hotmail.co.jp>

References

I. V. Oseledets, (2011). Tensor-Train Decomposition. *SIAM J. SCI. COMPUT.*

Yuan, Longhao, et. al., (2017). Completion of high order tensor data with missing entries via tensor-train decomposition. *International Conference on Neural Information Processing*

I. V. Oseledets, et. al., (2010). TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*

Ali Civril, et. al., (2009). On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*

See Also

TTSVD,TTWOPT,TTCross,skeleton.decomp,maxvol

Examples

```
ls("package:ttTensor")
```

---

maxvol — *maxvol algorithm*

Description

maxvol finds the r*r submatrix of maximal volume in C (n*r) by greedily searching the vector of max norm, and subtractction of its projection from the rest of rows. See also http://tensorly.org/stable/_modules/tensorly/contr

Usage

```
maxvol(C)
```

Arguments

C	The input sparse matrix.

Value

row_idx : The indices of rows, which make the determinant as large

Author(s)

Koki Tsuyuzaki

References

Ali Civril, et. al., (2009). On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*

See Also

skeleton.decomp

Examples

```
library("Matrix")
# Matrix data
X3 <- matrix(runif(10*20), nrow=10)
X3 <- as(X3, "sparseMatrix")
# Skeleton Decomposition
out.SKD <- skeleton.decomp(X3, r=3, num.iter=2, thr=1E-5)
```

---

skeleton.decomp *Skeleton Decomposition*

Description

skeleton.decomp decomposes the input sparse matrix (n*m) and return the three matrices C (n*r), U (r*r), and R (r*m). Only sparse matrix defined by the Matrix package is acceptable as the input.

Usage

```
skeleton.decomp(A, r, thr=1E-10, num.iter=30)
```

Arguments

A	The input sparse matrix.
r	Rank parameter to specify the lower dimension (r <= min(A)).
thr	The threshold to determine the convergence (Default: 1E-10).
num.iter	The number of iteration (Default: 30).

Value

C : A[I, :] U : inverse(A[I, J]) R : A[:, J] rowidx :The indices of rows colidx : The indices of columns RecError : The reconstruction error between data matrix and reconstructed matrix from C, U, and R RelChange : The relative change of the error

Author(s)

Koki Tsuyuzaki

References

I. V. Oseledets, et. al., (2010). TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*

See Also

maxvol

Examples

```
library("Matrix")
# Matrix data
X3 <- matrix(runif(10*20), nrow=10)
X3 <- as(X3, "sparseMatrix")
# Skeleton Decomposition
out.SKD <- skeleton.decomp(X3, r=3, num.iter=2, thr=1E-5)
```

TTCross *Tensor-Train Decomposition by TRCross*

Description

TTCross incrementaly decomposes the input tensor by skeleton decomposition algorithm. The algorithm only select the row/column indices and any large temporal matrix are genrated in the process. Therefore, this method is suitable for the sparse tensor.

Usage

```
TTCross(A, Ranks=NULL, thr=1E-10, num.iter=30)
```

Arguments

A	The input sparse tensor.
Ranks	TT-ranks to specify the lower dimensions.
thr	The threshold to determine the convergence (Default: 1E-10).
num.iter	The number of iteration (Default: 30).

Value

G : Core tensors

Author(s)

Koki Tsuyuzaki

References

I. V. Oseledets, et. al., (2010). TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*

Examples

```
library("rTensor")
library("tensorr")
# Sparse Tensor data
X1 <- array(rnorm(3*5*7*9*11), c(3,5,7,9,11))
dimnames(X1) <- list(
    I=paste0("i", 1:3),
    J=paste0("j", 1:5),
    K=paste0("k", 1:7),
    L=paste0("l", 1:9),
    M=paste0("m", 1:11)
    )
X1 <- as.tensor(X1)
X2 <- as_sptensor(dtensor(X1@data))
dimnames(X2) <- dimnames(X1@data)
# TT-ranks
Ranks <- c(p=2, q=4, r=6, s=8)
# TT-Cross
out.TTCross <- TTCross(X2, Ranks, num.iter=2)
```

`TTSVD` *Tensor-Train Decomposition by TTSVD*

Description

TTSVD incrementaly decomposes the input tensor by singular value decomposition (SVD).

Usage

```
TTSVD(A, Ranks=NULL, accuracy=NULL)
```

Arguments

`A`	The input tensor.
`Ranks`	TT-ranks to specify the lower dimensions.
`accuracy`	The accuracy of the compression.

Value

G : Core tensors

Author(s)

Koki Tsuyuzaki

References

I. V. Oseledets, (2011). Tensor-Train Decomposition. *SIAM J. SCI. COMPUT.*

Examples

```
library("rTensor")
# Tensor data
X1 <- array(rnorm(3*5*7*9*11), c(3,5,7,9,11))
dimnames(X1) <- list(
    I=paste0("i", 1:3),
    J=paste0("j", 1:5),
    K=paste0("k", 1:7),
    L=paste0("l", 1:9),
    M=paste0("m", 1:11)
    )
X1 <- as.tensor(X1)
# TT-ranks
Ranks <- c(p=2, q=4, r=6, s=8)
# TTSVD
out.TTSVD <- TTSVD(X1, Ranks)
out.TTSVD <- TTSVD(X1, accuracy=1E-10)
```

TTWOPT *Tensor-Train Decomposition by Tensor-train Weighted OPTimization*

Description

TTWOPT incrementaly decomposes the input tensor by gradient desecent. The tensor with missing entries is also specified with weight tensor W.

Usage

```
TTWOPT(X, Ranks, W=NULL, eta=1E-7, thr=1E-10, num.iter=100)
```

Arguments

X	The input tensor.
Ranks	TT-ranks to specify the lower dimensions.
W	The weight tensor to specify the missing entries (0: missing, 1: existing). The size must be same as that of X.
eta	The learning rate parameter of the gradient descent algorithm (Default : 1E-10).
thr	The threshold to determine the convergence (Default: 1E-10).
num.iter	The number of iteration (Default: 30).

Value

G : Core tensors RelChange : The relative change of the error f : The values of the object function RecError : The reconstruction error between data tensor and reconstructed tensor from C, U, and R

Author(s)

Koki Tsuyuzaki

References

Yuan, Longhao, et. al., (2017). Completion of high order tensor data with missing entries via tensor-train decomposition. *International Conference on Neural Information Processing*

Examples

```
library("rTensor")
# Tensor data
X1 <- array(rnorm(3*5*7*9*11), c(3,5,7,9,11))
dimnames(X1) <- list(
    I=paste0("i", 1:3),
    J=paste0("j", 1:5),
    K=paste0("k", 1:7),
    L=paste0("l", 1:9),
    M=paste0("m", 1:11)
    )
X1 <- as.tensor(X1)
# TT-ranks
Ranks <- c(p=2, q=4, r=6, s=8)
# TTWOPT
out.TTWOPT <- TTWOPT(X1, Ranks, eta=1E-7)
```

Index

∗Topic **methods**
- maxvol, 3
- skeleton.decomp, 4
- TTCross, 5
- TTSVD, 6
- TTWOPT, 7

∗Topic **package**
- ttTensor-package, 2

maxvol, *3*, 3, *4*

skeleton.decomp, *3*, 4

TTCross, *3*, 5
TTSVD, *3*, 6
ttTensor *(*ttTensor-package*)*, 2
ttTensor-package, 2
TTWOPT, *3*, 7