

# Package ‘tensorr’

January 21, 2019

**Title** Sparse Tensors in R

**Version** 0.1.1

**Description** Provides methods to manipulate and store sparse tensors. Tensors are multidimensional generalizations of matrices (two dimensional) and vectors (one dimensional).

**Depends** R (>= 3.3.0)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**URL** <https://github.com/zamorarr/tensorr>

**BugReports** <https://github.com/zamorarr/tensorr/issues>

**Imports** assertive.base, assertive.properties, assertive.types, Matrix, methods, purrr

**Suggests** covr, knitr, rmarkdown, testthat

**RoxygenNote** 6.1.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Robert Zamora [aut, cre]

**Maintainer** Robert Zamora <zamora.rr@gmail.com>

**Repository** CRAN

**Date/Publication** 2019-01-21 05:30:03 UTC

## R topics documented:

as_dtensor . . . . .	2
as_sptensor . . . . .	3
dim,dtensor-method . . . . .	3
dimnames,dtensor-method . . . . .	4
dtensor . . . . .	5

dtensor-class . . . . .	5
dtensor-extract . . . . .	6
dtensor-replace . . . . .	6
fill_missing_indices . . . . .	7
innerprod . . . . .	8
is_dtensor . . . . .	8
length,tensor-method . . . . .	9
norm . . . . .	9
nzsubs . . . . .	10
nzvals . . . . .	10
outerprod . . . . .	11
refold . . . . .	11
sptensor . . . . .	12
sptensor-class . . . . .	13
squeeze . . . . .	13
tensorr . . . . .	14
ttm . . . . .	14
ttv . . . . .	15
unfold . . . . .	16
unfolded_dtensor . . . . .	16
unfolded_dtensor-class . . . . .	17
unfolded_sptensor . . . . .	17
unfolded_sptensor-class . . . . .	18
[,sptensor,character,missing,ANY-method . . . . .	18
[<-,sptensor,character,missing,ANY-method . . . . .	19
<b>Index</b>	<b>21</b>

---

as_dtensor	<i>Convert objects to dense tensors</i>
------------	---

---

## Description

Convert objects to dense tensors

## Usage

```
as_dtensor(x, ...)
```

```
## S4 method for signature 'sptensor'
as_dtensor(x)
```

```
## S4 method for signature 'array'
as_dtensor(x)
```

**Arguments**

x	object
...	extra params

---

as_sptensor	<i>Convert objects to sparse tensors</i>
-------------	--

---

**Description**

Convert objects to sparse tensors

**Usage**

```
as_sptensor(x, ...)
```

```
## S4 method for signature 'sptensor'
```

```
as_sptensor(x)
```

```
## S4 method for signature 'dtensor'
```

```
as_sptensor(x)
```

```
## S4 method for signature 'data.frame'
```

```
as_sptensor(x, valcol = NULL, dims = NULL)
```

**Arguments**

x	object
...	extra params
valcol	column to use for the tensor values. all other columns are treated as indices
dims	dimensions of tensor. If not provided, the maximum value for each of the indices is used.

---

dim,dtensor-method	<i>Dimensions of a tensor</i>
--------------------	-------------------------------

---

**Description**

Dimensions of a tensor

**Usage**

```
## S4 method for signature 'dtensor'
```

```
dim(x)
```

```
## S4 method for signature 'sptensor'
```

```
dim(x)
```

**Arguments**

x                    tensor

---

dimnames,dtensor-method

*Dimension names of a tensor*

---

**Description**

Dimension names of a tensor

**Usage**

```
## S4 method for signature 'dtensor'
dimnames(x)

## S4 replacement method for signature 'dtensor,list'
dimnames(x) <- value

## S4 replacement method for signature 'dtensor,NULL'
dimnames(x) <- value

## S4 replacement method for signature 'dtensor,ANY'
dimnames(x) <- value

## S4 method for signature 'sptensor'
dimnames(x)

## S4 replacement method for signature 'sptensor,list'
dimnames(x) <- value

## S4 replacement method for signature 'sptensor,NULL'
dimnames(x) <- value

## S4 replacement method for signature 'sptensor,ANY'
dimnames(x) <- value
```

**Arguments**

x                    tensor

value                replacement value. Must be a list of length equal to the number of dimensions in the tensor

---

dtensor	<i>Construct a dense tensor</i>
---------	---------------------------------

---

**Description**

Construct a [dtensor](#) from a multi-dimensional array.

**Usage**

```
dtensor(x)

## S4 method for signature 'array'
dtensor(x)

## S4 method for signature 'numeric'
dtensor(x)
```

**Arguments**

x                    n-dimensional R array

**See Also**

[dtensor-class](#) for class documentation.

**Examples**

```
# A 2x2x2 dense tensor
arr <- array(data = c(1,0,0,0,1,0,0,0) , dim = c(2,2,2))
X <- dtensor(arr)
```

---

dtensor-class	<i>An S4 class for a dense tensor</i>
---------------	---------------------------------------

---

**Description**

Simple wrapper around an n-dimensional array.

**Slots**

x n dimensional array

---

dtensor-extract	<i>Extract values from a dense tensor</i>
-----------------	---

---

**Description**

Extract values from a dense tensor

**Usage**

```
## S4 method for signature 'dtensor,missing,missing,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'dtensor,numeric,missing,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'dtensor,missing,numeric,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'dtensor,numeric,numeric,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'dtensor,matrix,missing,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'dtensor,list,missing,ANY'
x[i, j, ..., drop = FALSE]
```

**Arguments**

x	dtensor object
i	numeric index, vector, list, or matrix
j	numeric index
...	additional numeric indices
drop	whether to drop dimensions

---

dtensor-replace	<i>Replace values from a dense tensor</i>
-----------------	---

---

**Description**

Replace values from a dense tensor

**Usage**

```
## S4 replacement method for signature 'dtensor,ANY,ANY,ANY'  
x[i, j, ...] <- value  
  
## S4 replacement method for signature 'dtensor,ANY,missing,ANY'  
x[i, j, ...] <- value  
  
## S4 replacement method for signature 'dtensor,matrix,missing,ANY'  
x[i, j, ...] <- value  
  
## S4 replacement method for signature 'dtensor,list,missing,ANY'  
x[i, j, ...] <- value
```

**Arguments**

x	dtensor object
i	numeric index, vector, list, or matrix
j	numeric index
...	additional numeric indices
value	replacement value(s)

---

fill\_missing\_indices *Fill NULL indices with a range from 1:dim*

---

**Description**

Fill NULL indices with a range from 1:dim

**Usage**

```
fill_missing_indices(index, dim)
```

**Arguments**

index	a numeric value or NULL
dim	size of dimension

---

innerprod	<i>Calculate the inner product of a pair of tensors</i>
-----------	---

---

**Description**

Calculate the inner product of a pair of tensors

**Usage**

```
innerprod(x, y)

## S4 method for signature 'dtensor,dtensor'
innerprod(x, y)

## S4 method for signature 'sptensor,sptensor'
innerprod(x, y)
```

**Arguments**

x, y	tensors
------	---------

---

is_dtensor	<i>Test if the object is a tensor</i>
------------	---------------------------------------

---

**Description**

Test if the object is a tensor

**Usage**

```
is_dtensor(x)

is_sptensor(x)

is_tensor(x)
```

**Arguments**

x	object
---	--------

**Functions**

- is\_dtensor: dense tensor
- is\_sptensor: sparse tensor



---

length, tensor-method    *Length of a tensor*

---

**Description**

The total number of values in a tensor. Equal to the product of the dimensions.

**Usage**

```
## S4 method for signature 'tensor'  
length(x)
```

**Arguments**

x                    tensor

---

norm                    *Calculate the Frobenius norm of a tensor*

---

**Description**

Calculate the Frobenius norm of a tensor

**Usage**

```
norm(x)  
  
## S4 method for signature 'dtensor'  
norm(x)  
  
## S4 method for signature 'sptensor'  
norm(x)
```

**Arguments**

x                    tensor

---

nzsubs	<i>Subscripts of non-zero values in a tensor</i>
--------	--

---

**Description**

Subscripts of non-zero values in a tensor

**Usage**

```
nzsubs(x)
```

```
## S4 method for signature 'dtensor'  
nzsubs(x)
```

```
## S4 method for signature 'sptensor'  
nzsubs(x)
```

**Arguments**

x	tensor
---	--------

---

nzvals	<i>Non-zero values in a tensor</i>
--------	------------------------------------

---

**Description**

Non-zero values in a tensor

**Usage**

```
nzvals(x)
```

```
## S4 method for signature 'dtensor'  
nzvals(x)
```

```
## S4 method for signature 'sptensor'  
nzvals(x)
```

**Arguments**

x	tensor
---	--------

---

outerprod	<i>Calculate the outer product of a pair of tensors</i>
-----------	---

---

**Description**

Calculate the outer product of a pair of tensors

**Usage**

```
outerprod(x, y)
```

```
ttt(x, y)
```

```
## S4 method for signature 'dtensor,dtensor'  
outerprod(x, y)
```

```
## S4 method for signature 'dtensor,tensor'  
outerprod(x, y)
```

```
## S4 method for signature 'tensor,dtensor'  
outerprod(x, y)
```

```
## S4 method for signature 'tensor,tensor'  
ttt(x, y)
```

```
## S4 method for signature 'sptensor,sptensor'  
outerprod(x, y)
```

**Arguments**

x, y	tensors
------	---------

---

refold	<i>Refold an unfolded tensor</i>
--------	----------------------------------

---

**Description**

Refold an unfolded tensor

**Usage**

```
refold(x)
```

```
## S4 method for signature 'unfolded_dtensor'  
refold(x)
```

```
## S4 method for signature 'unfolded_sptensor'
refold(x)
```

### Arguments

x                    an unfolded tensor

---

sptensor                    *Construct a sparse tensor*

---

### Description

Construct an [sptensor](#) from a matrix of subscripts for non-zero values, a vector of non-zero values, and the numeric dimensions of the tensor.

### Usage

```
sptensor(subs, vals, dims)

## S4 method for signature 'matrix,ANY,numeric'
sptensor(subs, vals, dims)

## S4 method for signature 'matrix,missing,numeric'
sptensor(subs, vals, dims)

## S4 method for signature 'list,ANY,numeric'
sptensor(subs, vals, dims)
```

### Arguments

subs                    matrix with length(dims) rows and length(vals) cols. Each row in the matrix corresponds to a different tensor dimension. Each column in the matrix represents a vector of subscripts pointing to a non-zero value in the tensor.

vals                    values of non-zero entries. The subscripts for the first value are the first column of the subs matrix, the subscripts for the second value are in the second column of the subs matrix, etc...

dims                    sizes of each dimension

### See Also

[sptensor-class](#) for class documentation.

**Examples**

```
# A 2x2x2 sparse tensor
subs <- matrix(c(1,1,1, 1,1,2), c(3,2))
vals <- c(10,20)
dims <- c(2,2,2)
X <- sptensor(subs, vals, dims)
```

---

sptensor-class	<i>An S4 class for a sparse tensor</i>
----------------	--

---

**Description**

Stores the tensor in co-ordinate (COO) format. Non-zero entries are stored by their subscripts (i1,i2,i3,...,in) the subs matrix and their vals in the vals vector.

**Slots**

subs matrix with length(dims) rows and length(vals) cols.

vals values of non-zero entries.

dims sizes of each dimension

---

squeeze	<i>Remove tensor dimensions of size 1</i>
---------	---

---

**Description**

Remove tensor dimensions of size 1

**Usage**

```
squeeze(x, todrop = NULL)
```

**Arguments**

x	sptensor
todrop	specific dimensions to drop. If NULL, will drop all dimensions of size 1

---

tenso	<i>tenso: sparse tensors in R</i>
-------	-----------------------------------

---

### Description

tenso provides methods to manipulate and store sparse tensors. Tensors are multi-dimensional generalizations of matrices (two dimensional) and vectors (one dimensional).

### Details

It has three main goals:

- Provide an efficient format to store sparse tensors in R.
- Provide standard tensor operations such as multiplication and unfolding.
- Provide standard tensor decomposition techniques such as CP and Tucker.

### References

Many of the dense and sparse implementation ideas were adapted from

- *B. W. Bader and T. G. Kolda. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping, ACM Transactions on Mathematical Software 32(4):635-653, December 2006.*
- *B. W. Bader and T. G. Kolda. Efficient MATLAB computations with sparse and factored tensors, SIAM Journal on Scientific Computing 30(1):205-231, December 2007.*

For a review on tensors, see

- *T. G. Kolda and B. W. Bader, Tensor Decompositions and Applications, SIAM Review 51(3):455-500, September 2009*

---

ttm	<i>Tensor times matrix</i>
-----	----------------------------

---

### Description

Calculates the n-mode product of a tensor and a matrix. Given a tensor X with dimensions  $I_1, I_2, \dots, I_n, I_n + 1, \dots, I_N$  and a matrix U with dimensions  $J, I_n$ , the resulting tensor after multiplication will have dimension  $I_1, I_2, \dots, J, I_n + 1, \dots, I_N$ .

**Usage**

```

ttm(x, u, mode)

## S4 method for signature 'dtensor,Matrix,numeric'
ttm(x, u, mode)

## S4 method for signature 'dtensor,matrix,numeric'
ttm(x, u, mode)

## S4 method for signature 'sptensor,Matrix,numeric'
ttm(x, u, mode)

## S4 method for signature 'sptensor,matrix,numeric'
ttm(x, u, mode)

```

**Arguments**

x	tensor
u	matrix
mode	mode along tensor to perform multiplication

**References**

*T. G. Kolda and B. W. Bader, Tensor Decompositions and Applications, SIAM Review 51(3):455-500, September 2009*

---

ttv

*Tensor times vector*


---

**Description**

Calculates the n-mode product of a tensor and a vector Given a tensor X with dimensions  $I_1, I_2, \dots, I_n, I_n + 1, \dots, I_N$  and a vector v with dimensions  $J, 1$ , the resulting tensor after multiplication will have dimension  $I_1, I_2, \dots, I_n - 1, I_n + 1, \dots, I_N$ . Note that the dimension corresponding to the mode has been dropped.

**Usage**

```

ttv(x, v, mode)

## S4 method for signature 'dtensor,numeric,numeric'
ttv(x, v, mode)

## S4 method for signature 'sptensor,numeric,numeric'
ttv(x, v, mode)

## S4 method for signature 'sptensor,sparseVector,numeric'
ttv(x, v, mode)

```

**Arguments**

x	tensor
v	vector
mode	mode along tensor to perform multiplication

**References**

*T. G. Kolda and B. W. Bader, Tensor Decompositions and Applications, SIAM Review 51(3):455-500, September 2009*

---

unfold	<i>Unfold (matricize) a tensor along a mode</i>
--------	---

---

**Description**

Unfold (matricize) a tensor along a mode

**Usage**

```
unfold(x, mode)

## S4 method for signature 'dtensor,numeric'
unfold(x, mode)

## S4 method for signature 'sptensor,numeric'
unfold(x, mode)
```

**Arguments**

x	tensor
mode	dimension to unfold along

---

unfolded_dtensor	<i>Construct an unfolded dense tensor</i>
------------------	---

---

**Description**

Construct an [unfolded\\_dtensor](#) from a dense matrix, mode, and dimensions of original tensor.

**Usage**

```
unfolded_dtensor(mat, mode, tensor_dims)

## S4 method for signature 'Matrix,numeric,numeric'
unfolded_dtensor(mat, mode, tensor_dims)
```



**Arguments**

mat	dense matrix representing unfolded tensor
mode	dimension to unfold tensor along
tensor_dims	original dimensions of tensor (useful for re-folding)

**See Also**

[unfolding\\_dtensor-class](#) for class documentation.

---

unfolding\_dtensor-class

*An S4 class for an unfolded dense tensor*

---

**Description**

An unfolded ("matricized") tensor along a specified dimension. Can be easily refolded back into a tensor with command `refold`

**Slots**

mat	sparse matrix representing unfolded tensor
mode	dimension along which tensor was unfolded
tensor_dims	dimensions of original tensor

---

unfolding\_sptensor

*Construct an unfolded sparse tensor*

---

**Description**

Construct an [unfolding\\_sptensor](#) from a sparse matrix, mode, and dimensions of original tensor.

**Usage**

```
unfolding_sptensor(mat, mode, tensor_dims)

## S4 method for signature 'Matrix,numeric,numeric'
unfolding_sptensor(mat, mode,
  tensor_dims)
```

**Arguments**

mat	sparse matrix representing unfolded tensor
mode	dimension to unfold tensor along
tensor_dims	original dimensions of tensor (useful for re-folding)

**See Also**

[unfolded\\_sptensor-class](#) for class documentation.

---

unfolded\_sptensor-class

*An S4 class for an unfolded sparse tensor*

---

**Description**

An unfolded ("matricized") tensor along a specified dimension. Can be easily refolded back into a tensor with command `refold`

**Slots**

`mat` sparse matrix representing unfolded tensor

`mode` dimension along which tensor was unfolded

`tensor_dims` dimensions of original tensor

---

[,sptensor,character,missing,ANY-method

*Extract values from a sparse tensor*

---

**Description**

There are multiple ways to provide indices for a sparse tensor. For example, if you have a three-dimensional tensor you can provide indices separated by a comma or a numeric vector of linear indices:

- `c(x[1,1,1], x[1,1,2])`
- `x[c(1, 5)]`

However for tensors with high dimensions this can be cumbersome to write. Therefore the tensor class also allows you to extract values with indices in matrix or list form, which are more suited for non-interactive coding.

- `x[matrix(c(1,1,1,1,1,2), nrow = 3)]`
- `x[list(c(1,1,1), c(1,1,2))]`

**Usage**

```
## S4 method for signature 'sptensor,character,missing,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'sptensor,missing,character,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'sptensor,character,character,ANY'
x[i, j, ...,
  drop = FALSE]

## S4 method for signature 'sptensor,missing,missing,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'sptensor,numeric,missing,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'sptensor,missing,numeric,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'sptensor,numeric,numeric,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'sptensor,list,missing,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'sptensor,matrix,missing,ANY'
x[i, j, ..., drop = FALSE]
```

**Arguments**

x	sptensor object
i	numeric index, vector, list, or matrix
j	numeric index
...	additional numeric indices
drop	whether to drop dimensions

---

```
[<- ,sptensor,character,missing,ANY-method
  Replace values from a sparse tensor
```

---

**Description**

Replace values from a sparse tensor

**Usage**

```
## S4 replacement method for signature 'sptensor,character,missing,ANY'  
x[i, j, ...] <- value  
  
## S4 replacement method for signature 'sptensor,missing,character,ANY'  
x[i, j, ...] <- value  
  
## S4 replacement method for signature 'sptensor,character,character,ANY'  
x[i, j, ...] <- value  
  
## S4 replacement method for signature 'sptensor,missing,missing,ANY'  
x[i, j, ...] <- value  
  
## S4 replacement method for signature 'sptensor,numeric,missing,ANY'  
x[i, j, ...] <- value  
  
## S4 replacement method for signature 'sptensor,missing,numeric,ANY'  
x[i, j, ...] <- value  
  
## S4 replacement method for signature 'sptensor,numeric,numeric,ANY'  
x[i, j, ...] <- value  
  
## S4 replacement method for signature 'sptensor,list,missing,ANY'  
x[i, j, ...] <- value  
  
## S4 replacement method for signature 'sptensor,matrix,missing,ANY'  
x[i, j, ...] <- value
```

**Arguments**

x	sptensor object
i	numeric index, vector, list, or matrix
j	numeric index
...	additional numeric indices
value	replacement value(s)

# Index

- [,dtensor,list,missing,ANY-method  
(dtensor-extract), 6
- [,dtensor,list,missing-method  
(dtensor-extract), 6
- [,dtensor,matrix,missing,ANY-method  
(dtensor-extract), 6
- [,dtensor,matrix,missing-method  
(dtensor-extract), 6
- [,dtensor,missing,missing,ANY-method  
(dtensor-extract), 6
- [,dtensor,missing,missing-method  
(dtensor-extract), 6
- [,dtensor,missing,numeric,ANY-method  
(dtensor-extract), 6
- [,dtensor,missing,numeric-method  
(dtensor-extract), 6
- [,dtensor,numeric,missing,ANY-method  
(dtensor-extract), 6
- [,dtensor,numeric,missing-method  
(dtensor-extract), 6
- [,dtensor,numeric,numeric,ANY-method  
(dtensor-extract), 6
- [,dtensor,numeric,numeric-method  
(dtensor-extract), 6
- [,sptensor,character,character,ANY-method  
([,sptensor,character,missing,ANY-method),  
18
- [,sptensor,character,missing,ANY-method,  
18
- [,sptensor,list,missing,ANY-method  
([,sptensor,character,missing,ANY-method),  
18
- [,sptensor,list,missing-method  
([,sptensor,character,missing,ANY-method),  
18
- [,sptensor,matrix,missing,ANY-method  
([,sptensor,character,missing,ANY-method),  
18
- [,sptensor,matrix,missing-method  
([,sptensor,character,missing,ANY-method),  
18
- ([,sptensor,character,missing,ANY-method),  
18
- [,sptensor,missing,character,ANY-method  
([,sptensor,character,missing,ANY-method),  
18
- [,sptensor,missing,missing,ANY-method  
([,sptensor,character,missing,ANY-method),  
18
- [,sptensor,missing,missing-method  
([,sptensor,character,missing,ANY-method),  
18
- [,sptensor,missing,numeric,ANY-method  
([,sptensor,character,missing,ANY-method),  
18
- [,sptensor,missing,numeric-method  
([,sptensor,character,missing,ANY-method),  
18
- [,sptensor,numeric,missing,ANY-method  
([,sptensor,character,missing,ANY-method),  
18
- [,sptensor,numeric,missing-method  
([,sptensor,character,missing,ANY-method),  
18
- [,sptensor,numeric,numeric,ANY-method  
([,sptensor,character,missing,ANY-method),  
18
- [,sptensor,numeric,numeric-method  
([,sptensor,character,missing,ANY-method),  
18
- [<- ,sptensor,character,missing,ANY-method,  
19
- [<- ,dtensor,ANY,ANY,ANY-method  
(dtensor-replace), 6
- [<- ,dtensor,ANY,ANY-method  
(dtensor-replace), 6
- [<- ,dtensor,ANY,missing,ANY-method  
(dtensor-replace), 6
- [<- ,dtensor,ANY,missing-method  
(dtensor-replace), 6

[`<-`, dtensor, list, missing, ANY-method  
     (dtensor-replace), 6  
 [`<-`, dtensor, list, missing-method  
     (dtensor-replace), 6  
 [`<-`, dtensor, matrix, missing, ANY-method  
     (dtensor-replace), 6  
 [`<-`, dtensor, matrix, missing-method  
     (dtensor-replace), 6  
 [`<-`, sptensor, character, character, ANY-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19  
 [`<-`, sptensor, list, missing, ANY-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19  
 [`<-`, sptensor, list, missing-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19  
 [`<-`, sptensor, matrix, missing, ANY-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19  
 [`<-`, sptensor, matrix, missing-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19  
 [`<-`, sptensor, missing, character, ANY-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19  
 [`<-`, sptensor, missing, missing, ANY-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19  
 [`<-`, sptensor, missing, missing-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19  
 [`<-`, sptensor, missing, numeric, ANY-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19  
 [`<-`, sptensor, missing, numeric-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19  
 [`<-`, sptensor, numeric, missing, ANY-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19  
 [`<-`, sptensor, numeric, missing-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19  
 [`<-`, sptensor, numeric, numeric, ANY-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19  
 [`<-`, sptensor, numeric, numeric-method  
     ([`<-`, sptensor, character, missing, ANY-method),  
     19

([`<-`, sptensor, character, missing, ANY-method),  
 19  
 as\_dtensor, 2  
 as\_dtensor, array-method (as\_dtensor), 2  
 as\_dtensor, sptensor-method  
     (as\_dtensor), 2  
 as\_sptensor, 3  
 as\_sptensor, data.frame-method  
     (as\_sptensor), 3  
 as\_sptensor, dtensor-method  
     (as\_sptensor), 3  
 as\_sptensor, list-method  
     (as\_sptensor), 3  
 as\_sptensor, sptensor-method  
     (as\_sptensor), 3  
 dim (dim, dtensor-method), 3  
 dim, dtensor-method, 3  
 dim, sptensor-method  
     (dim, dtensor-method), 3  
 dimnames (dimnames, dtensor-method), 4  
 dimnames, dtensor-method, 4  
 dimnames, sptensor-method  
     (dimnames, dtensor-method), 4  
 dimnames<-, dtensor, ANY-method  
     (dimnames, dtensor-method), 4  
 dimnames<-, dtensor, list-method  
     (dimnames, dtensor-method), 4  
 dimnames<-, dtensor, NULL-method  
     (dimnames, dtensor-method), 4  
 dimnames<-, dtensor-method  
     (dimnames, dtensor-method), 4  
 dimnames<-, dtensor-method, ANY  
     (dimnames, dtensor-method), 4  
 dimnames<-, dtensor-method, NULL  
     (dimnames, dtensor-method), 4  
 dimnames<-, sptensor, ANY-method  
     (dimnames, dtensor-method), 4  
 dimnames<-, sptensor, list-method  
     (dimnames, dtensor-method), 4  
 dimnames<-, sptensor, NULL-method  
     (dimnames, dtensor-method), 4  
 dimnames<-, sptensor-method, ANY  
     (dimnames, dtensor-method), 4  
 dimnames<-, sptensor-method, list  
     (dimnames, dtensor-method), 4  
 dimnames<-, sptensor-method, NULL  
     (dimnames, dtensor-method), 4  
 dtensor, 5, 5  
 dtensor, array-method (dtensor), 5

- dtensor, numeric-method (dtensor), [5](#)
- dtensor-class, [5](#)
- dtensor-extract, [6](#)
- dtensor-replace, [6](#)
- fill\_missing\_indices, [7](#)
- innerprod, [8](#)
- innerprod, dtensor, dtensor-method (innerprod), [8](#)
- innerprod, sptensor, sptensor-method (innerprod), [8](#)
- is\_dtensor, [8](#)
- is\_sptensor (is\_dtensor), [8](#)
- is\_tensor (is\_dtensor), [8](#)
- length, tensor-method, [9](#)
- norm, [9](#)
- norm, dtensor-method (norm), [9](#)
- norm, sptensor-method (norm), [9](#)
- nzsubs, [10](#)
- nzsubs, dtensor-method (nzsubs), [10](#)
- nzsubs, sptensor-method (nzsubs), [10](#)
- nzvals, [10](#)
- nzvals, dtensor-method (nzvals), [10](#)
- nzvals, sptensor-method (nzvals), [10](#)
- outerprod, [11](#)
- outerprod, dtensor, dtensor-method (outerprod), [11](#)
- outerprod, dtensor, tensor-method (outerprod), [11](#)
- outerprod, sptensor, sptensor-method (outerprod), [11](#)
- outerprod, tensor, dtensor-method (outerprod), [11](#)
- refold, [11](#)
- refold, unfolded\_dtensor-method (refold), [11](#)
- refold, unfolded\_sptensor-method (refold), [11](#)
- sptensor, [12](#), [12](#)
- sptensor, list, ANY, numeric-method (sptensor), [12](#)
- sptensor, matrix, ANY, numeric-method (sptensor), [12](#)
- sptensor, matrix, missing, numeric-method (sptensor), [12](#)
- sptensor-class, [13](#)
- sptensor-extract ([, sptensor, character, missing, ANY-method), [18](#)
- sptensor-replace ([<- , sptensor, character, missing, ANY-method), [19](#)
- squeeze, [13](#)
- tensorr, [14](#)
- tensorr-package (tensorr), [14](#)
- ttn, [14](#)
- ttn, dtensor, Matrix, numeric, numeric-method (ttn), [14](#)
- ttn, dtensor, matrix, numeric, numeric-method (ttn), [14](#)
- ttn, dtensor, Matrix, numeric-method (ttn), [14](#)
- ttn, dtensor, matrix, numeric-method (ttn), [14](#)
- ttn, sptensor, Matrix, numeric, numeric-method (ttn), [14](#)
- ttn, sptensor, matrix, numeric, numeric-method (ttn), [14](#)
- ttn, sptensor, Matrix, numeric-method (ttn), [14](#)
- ttn, sptensor, matrix, numeric-method (ttn), [14](#)
- ttt (outerprod), [11](#)
- ttt, tensor, tensor-method (outerprod), [11](#)
- ttv, [15](#)
- ttv, dtensor, numeric, numeric, numeric-method (ttv), [15](#)
- ttv, dtensor, numeric, numeric-method (ttv), [15](#)
- ttv, sptensor, numeric, numeric, numeric-method (ttv), [15](#)
- ttv, sptensor, numeric, numeric-method (ttv), [15](#)
- ttv, sptensor, sparseVector, numeric-method (ttv), [15](#)
- unfold, [16](#)
- unfold, dtensor, numeric-method (unfold), [16](#)
- unfold, sptensor, numeric-method (unfold), [16](#)

unfolded\_dtensor, [16](#), [16](#)  
unfolded\_dtensor, Matrix, numeric, numeric-method  
    (unfolded\_dtensor), [16](#)  
unfolded\_dtensor-class, [17](#)  
unfolded\_sptensor, [17](#), [17](#)  
unfolded\_sptensor, Matrix, numeric, numeric-method  
    (unfolded\_sptensor), [17](#)  
unfolded\_sptensor, sparseMatrix, numeric, numeric-method  
    (unfolded\_sptensor), [17](#)  
unfolded\_sptensor-class, [18](#)