

Package ‘slam’

January 2, 2012

Version 0.1-23

Date 2011-10-27

Title Sparse Lightweight Arrays and Matrices

Description Data structures and algorithms for sparse arrays and matrices, based on index arrays and simple triplet representations, respectively.

Author Kurt Hornik, David Meyer, Christian Buchta

Maintainer David Meyer <David.Meyer@R-project.org>

Depends R (>= 2.8.0)

Enhances Matrix, SparseM, spam

License GPL-2

Repository CRAN

Date/Publication 2011-10-27 11:10:23

R topics documented:

crossprod	2
foreign	3
rollup	3
simple_sparse_array	5
simple_triplet_matrix	6
sums	7

Index	9
--------------	----------

`crossprod`*Matrix Crossproduct*

Description

Compute the matrix cross-product of a sparse and dense matrix.

Usage

```
tcrossprod_simple_triplet_matrix(x, y = NULL)
```

Arguments

`x` a matrix in `simple_triplet_matrix`-form.
`y` a numeric matrix.

Details

Provides fast computation of `x %*% t(x)` and `x %*% t(y)` (`tcrossprod`).

Value

A double matrix, with appropriate `dimnames` taken from `x` and `y`.

Note

If `y` (or `x`) contains any of the special values `NA`, `NaN`, or `Inf` (if `y = NULL`) then `x` is coerced to `matrix` and the computation is delegated to `tcrossprod`.

Author(s)

Christian Buchta

See Also

[crossprod](#) for dense-on-dense computations.

Examples

```
##  
x <- matrix(c(1, 0, 0, 2, 1, 0), nrow = 3)  
x  
s <- as.simple_triplet_matrix(x)  
tcrossprod_simple_triplet_matrix(s, x)  
##  
tcrossprod_simple_triplet_matrix(s)
```

foreign

Read and Write Sparse Matrix Format Files

Description

Read and write CLUTO sparse matrix format files.

Usage

```
read_stm_CLUTO(file)
write_stm_CLUTO(x, file)
```

Arguments

file	a character string with the name of the file to read or write.
x	a matrix object.

Details

Documentation for CLUTO including its sparse matrix format is available from <http://www-users.cs.umn.edu/~karypis/cluto/>.

read_stm_CLUTO reads CLUTO sparse matrices, returning a [simple triplet matrix](#).

write_stm_CLUTO writes CLUTO sparse matrices. Argument x must be coercible to a simple triplet matrix via [as.simple_triplet_matrix](#).

rollup

Rollup Sparse Arrays

Description

Rollup (aggregate) sparse arrays along arbitrary dimensions.

Usage

```
rollup(x, MARGIN, INDEX, FUN, ...)
```

```
## S3 method for class 'simple_triplet_matrix'
rollup(x, MARGIN, INDEX, FUN = sum, ...)
## S3 method for class 'simple_sparse_array'
rollup(x, MARGIN, INDEX, FUN = sum, ...)
## S3 method for class 'matrix'
rollup(x, MARGIN, INDEX, FUN = sum, ...)
## S3 method for class 'array'
rollup(x, MARGIN, INDEX, FUN = sum, ...)
```

Arguments

<code>x</code>	a sparse array containing numeric, integer, or logical values.
<code>MARGIN</code>	a vector giving the subscripts (names) of the dimensions to be rolled up.
<code>INDEX</code>	a corresponding (list of) factor (components) in the sense that <code>as.factor(f)</code> defines the grouping.
<code>FUN</code>	the name of the function to be applied.
<code>...</code>	optional arguments to <code>FUN</code> .

Details

Provides fast summation over the rows or columns of sparse matrices in `simple_triplet`-form.

Note that the default method tries to coerce `x` to array.

Value

An object of the same class as `x` where for class `simple_triplet_matrix` the values are always of type `double`. The `dimnames` are taken from (the components of) `INDEX`.

Note

If (a component of) `INDEX` contains `NA` values the corresponding positions are ignored.

Author(s)

Christian Buchta

See Also

`simple_triplet_matrix` and `simple_sparse_array` for sparse arrays.

Examples

```
##
x <- matrix(c(1, 0, 0, 2, 1, NA), nrow = 2,
            dimnames = list(A = 1:2, B = 1:3))
x
zz <- rollup(x, 2L, c(1,2,1), na.rm = TRUE)
zz
##
s <- as.simple_triplet_matrix(x)
z <- rollup(s, 2L, c(1,2,1), na.rm = TRUE)
identical(zz, as.matrix(z))
##
a <- as.simple_sparse_array(s)
z <- rollup(a, 2L, c(1,2,1), na.rm = TRUE)
identical(as.array(zz), as.array(z))
##
zz <- rollup(x, 2L, c(1,NA,1), na.rm = TRUE)
zz
```

```
##
z <- rollup(s, 2L, c(1,NA,1), na.rm = TRUE)
identical(zz, as.matrix(z))
##
z <- rollup(a, 2L, c(1,NA,1), na.rm = TRUE)
identical(as.array(zz), as.array(z))
```

simple_sparse_array *Simple Sparse Arrays*

Description

Data structures and operators for sparse arrays based on a representation by index matrix and value vector.

Usage

```
simple_sparse_array(i, v, dim = NULL, dimnames = NULL)
```

```
as.simple_sparse_array(x)
```

```
is.simple_sparse_array(x)
```

Arguments

<code>i</code>	Integer matrix of array indices.
<code>v</code>	Vector of values.
<code>dim</code>	Integer vector specifying the size of the dimensions.
<code>dimnames</code>	either NULL or the names for the dimensions. This is a list with one component for each dimension, either NULL or a character vector of the length given by <code>dim</code> for that dimension. The list can be named, and the list names will be used as names for the dimensions. If the list is shorter than the number of dimensions, it is extended by NULL's to the length required.
<code>x</code>	An R object.

Details

`simple_sparse_array` is a generator for a class of “lightweight” sparse arrays, represented by index matrices and value vectors. Currently, only methods for indexing and coercion are implemented.

See Also

[simple_sparse_array](#) for sparse arrays.

Examples

```
x <- array(c(1, 0, 0, 2, 0, 0, 0, 3), dim = c(2, 2, 2))
s <- as.simple_sparse_array(x)
identical(x, as.array(s))

simple_sparse_array(matrix(c(1, 1, 1, 3, 3, 3), nrow = 2), c(1, 2))
```

simple_triplet_matrix *Simple Triplet Matrix*

Description

Data structures and operators for sparse matrices based on simple triplet representation.

Usage

```
simple_triplet_matrix(i, j, v, nrow = max(i), ncol = max(j), dimnames = NULL)
simple_triplet_zero_matrix(nrow, ncol = nrow, mode = "double")
simple_triplet_diag_matrix(v, nrow = length(v))

as.simple_triplet_matrix(x)
is.simple_triplet_matrix(x)
```

Arguments

<code>i, j</code>	Integer vectors of row and column indices, respectively.
<code>v</code>	Vector of values.
<code>nrow, ncol</code>	Integer values specifying the number of rows and columns, respectively. Defaults are the maximum row and column indices, respectively.
<code>dimnames</code>	A <code>dimnames</code> attribute for the matrix: <code>NULL</code> or a list of length 2 giving the row and column names respectively. An empty list is treated as <code>NULL</code> , and a list of length one as row names. The list can be named, and the list names will be used as names for the dimensions.
<code>mode</code>	Character string specifying the mode of the values.
<code>x</code>	An R object.

Details

`simple_triplet_matrix` is a generator for a class of “lightweight” sparse matrices, “simply” represented by triplets (`i`, `j`, `v`) of row indices `i`, column indices `j`, and values `v`, respectively. `simple_triplet_zero_matrix` and `simple_triplet_diag_matrix` are convenience functions for the creation of empty and diagonal matrices.

Currently implemented operations include the addition, subtraction, multiplication and division of compatible simple triplet matrices, as well as the multiplication and division of a simple triplet matrix and a vector. Comparisons of the elements of a simple triplet matrices with a number are also provided. In addition, methods for indexing, combining by rows (`rbind`) and columns (`cbind`), transposing (`t`), concatenating (`c`), and detecting/extracting duplicated and unique rows are implemented.

See Also

[simple_sparse_array](#) for sparse arrays.

Examples

```
x <- matrix(c(1, 0, 0, 2), nrow = 2)
s <- as.simple_triplet_matrix(x)
identical(x, as.matrix(s))

simple_triplet_matrix(c(1, 4), c(1, 2), c(1, 2))
simple_triplet_zero_matrix(3)
simple_triplet_diag_matrix(1:3)

cbind(rbind(x, t(x)), rbind(x, x))
```

sums

Form Row and Column Sums and Means

Description

Form row and column sums and means for sparse arrays (currently `simple_triplet_matrix` only).

Usage

```
row_sums(x, na.rm = FALSE, dims = 1, ...)
col_sums(x, na.rm = FALSE, dims = 1, ...)
row_means(x, na.rm = FALSE, dims = 1, ...)
col_means(x, na.rm = FALSE, dims = 1, ...)

## S3 method for class 'simple_triplet_matrix'
row_sums(x, na.rm = FALSE, dims = 1, ...)
## S3 method for class 'simple_triplet_matrix'
col_sums(x, na.rm = FALSE, dims = 1, ...)
## S3 method for class 'simple_triplet_matrix'
row_means(x, na.rm = FALSE, dims = 1, ...)
## S3 method for class 'simple_triplet_matrix'
col_means(x, na.rm = FALSE, dims = 1, ...)
```

Arguments

<code>x</code>	a sparse array containing numeric, integer, or logical values.
<code>na.rm</code>	logical. Should missing values (including NaN) be omitted from the calculations?
<code>dims</code>	currently not used for sparse arrays.
<code>...</code>	currently not used for sparse arrays.

Details

Provides fast summation over the rows or columns of sparse matrices in `simple_triplet`-form.

Value

A numeric (double) array of suitable size, or a vector if the result is one-dimensional. The dimnames (or names for a vector result) are taken from the original array.

Note

Results are always of storage type double to avoid (integer) overflows.

Author(s)

Christian Buchta

See Also

`simple_triplet_matrix`, `colSums` for dense numeric arrays.

Examples

```
##  
x <- matrix(c(1, 0, 0, 2, 1, NA), nrow = 3)  
x  
s <- as.simple_triplet_matrix(x)  
row_sums(s)  
row_sums(s, na.rm = TRUE)  
col_sums(s)  
col_sums(s, na.rm = TRUE)
```

Index

*Topic **IO**

foreign, 3

*Topic **algebra**

crossprod, 2

rollup, 3

sums, 7

*Topic **arith**

rollup, 3

sums, 7

*Topic **array**

crossprod, 2

rollup, 3

sums, 7

*Topic **math**

simple_sparse_array, 5

simple_triplet_matrix, 6

as.simple_sparse_array
(simple_sparse_array), 5

as.simple_triplet_matrix, 3

as.simple_triplet_matrix
(simple_triplet_matrix), 6

col_means (sums), 7

col_sums (sums), 7

colSums, 8

crossprod, 2, 2

foreign, 3

is.simple_sparse_array
(simple_sparse_array), 5

is.simple_triplet_matrix
(simple_triplet_matrix), 6

read_stm_CLUTO (foreign), 3

rollup, 3

row_means (sums), 7

row_sums (sums), 7

simple triplet matrix, 3

simple_sparse_array, 5, 5, 7

simple_triplet_diag_matrix
(simple_triplet_matrix), 6

simple_triplet_matrix, 6

simple_triplet_zero_matrix
(simple_triplet_matrix), 6

sums, 7

tcrossprod_simple_triplet_matrix
(crossprod), 2

write_stm_CLUTO (foreign), 3