

# Package ‘coop’

August 29, 2016

**Type** Package

**Title** Co-Operation: Fast Covariance, Correlation, and Cosine Similarity Operations

**Version** 0.4-0

**Description** Fast implementations of the co-operations: covariance, correlation, and cosine similarity. The implementations are fast and memory-efficient and their use is resolved automatically based on the input data, handled by R's S3 methods. Full descriptions of the algorithms and benchmarks are available in the package vignettes.

**License** BSD 2-clause License + file LICENSE

**Depends** R (>= 3.1.0)

**Enhances** slam (>= 0.1.32), Matrix

**Suggests** memuse, knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** yes

**ByteCompile** yes

**URL** <https://github.com/wrathematics/coop>

**BugReports** <https://github.com/wrathematics/coop/issues>

**Maintainer** Drew Schmidt <wrathematics@gmail.com>

**RoxygenNote** 5.0.1

**Author** Drew Schmidt [aut, cre],  
Christian Heckendorf [ctb] (Caught some memory errors.)

**Repository** CRAN

**Date/Publication** 2016-04-05 08:14:28

## R topics documented:

coop-package . . . . .	2
cosine . . . . .	3
covar . . . . .	4
pcor . . . . .	5
sparsity . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

coop-package	<i>Cooperation: A Package of Co-Operations</i>
--------------	--

---

### Description

Fast implementations of the co-operations: covariance, correlation, and cosine similarity. The implementations are fast and memory-efficient and their use is resolved automatically based on the input data, handled by R's S3 methods. Full descriptions of the algorithms and benchmarks are available in the package vignettes.

Covariance and correlation should largely need no introduction. Cosine similarity is commonly needed in, for example, natural language processing, where the cosine similarity coefficients of all columns of a term-document or document-term matrix is needed.

### Implementation Details

Multiple storage schemes for the input data are accepted. For dense matrices, an ordinary R matrix input is accepted. For sparse matrices, a matrix in COO format, namely `simple_triplet_matrix` from the `slam` package, is accepted.

The implementation for dense matrix inputs is dominated by a symmetric rank-k update via the BLAS subroutine `dsyrk`; see the package vignette for a discussion of the algorithm implementation and complexity.

The implementation for two dense vector inputs is dominated by the product  $t(x) \%*\% y$  performed by the BLAS subroutine `dgemm` and the normalizing products  $t(y) \%*\% y$ , each computed via the BLAS function `dsyrk`.

### Author(s)

Drew Schmidt

---

`cosine`*Cosine Similarity*

---

**Description**

Compute the cosine similarity matrix efficiently. The function syntax and behavior is largely modeled after that of the `cosine()` function from the `lsa` package, although with a very different implementation.

**Usage**

```
cosine(x, y, use = "everything")
```

**Arguments**

<code>x</code>	A numeric matrix or vector.
<code>y</code>	A vector (when <code>x</code> is a vector) or missing (blank) when <code>x</code> is a matrix.
<code>use</code>	The NA handler, as in R's <code>cov()</code> and <code>cor()</code> functions. Options are "everything", "all.obs", and "complete.obs".

**Details**

See `?coop-package` for implementation details.

**Value**

The  $n \times n$  matrix of all pair-wise vector cosine similarities of the columns.

**Author(s)**

Drew Schmidt

**See Also**

[sparsity](#)

**Examples**

```
x <- matrix(rnorm(10*3), 10, 3)

coop::cosine(x)
coop::cosine(x[, 1], x[, 2])
```

---

`covar`*Covariance*

---

**Description**

An optimized, efficient implementation for computing covariance.

**Usage**

```
covar(x, y, use = "everything")
```

**Arguments**

<code>x</code>	A numeric matrix or vector.
<code>y</code>	A vector (when <code>x</code> is a vector) or missing (blank) when <code>x</code> is a matrix.
<code>use</code>	The NA handler, as in R's <code>cov()</code> and <code>cor()</code> functions. Options are "everything", "all.obs", and "complete.obs".

**Details**

See `?coop-package` for implementation details.

**Value**

The covariance matrix.

**Author(s)**

Drew Schmidt

**See Also**

[cosine](#)

**Examples**

```
x <- matrix(rnorm(10*3), 10, 3)
coop::pcor(x)
coop::pcor(x[, 1], x[, 2])
```

---

pcor

*Pearson Correlation*

---

## Description

An optimized, efficient implementation for computing the pearson correlation.

## Usage

```
pcor(x, y, use = "everything")
```

## Arguments

x	A numeric matrix or vector.
y	A vector (when x is a vector) or missing (blank) when x is a matrix.
use	The NA handler, as in R's <code>cov()</code> and <code>cor()</code> functions. Options are "everything", "all.obs", and "complete.obs".

## Details

See `?coop` for implementation details.

## Value

The pearson correlation matrix.

## Author(s)

Drew Schmidt

## See Also

[cosine](#)

## Examples

```
x <- matrix(rnorm(10*3), 10, 3)

coop::pcor(x)
coop::pcor(x[, 1], x[, 2])
```

---

sparsity	<i>Sparsity</i>
----------	-----------------

---

**Description**

Show the sparsity (as a count or proportion) of a matrix. For example, .99 sparsity means 99% of the values are zero. Similarly, a sparsity of 0 means the matrix is fully dense.

**Usage**

```
sparsity(x, proportion = TRUE)
```

**Arguments**

x	The matrix, stored as an ordinary R matrix or as a "simple triplet matrix" (from the slam package).
proportion	Logical; should a proportion or a count be returned?

**Details**

The implementation is very efficient for dense matrices. For sparse triplet matrices, the count is trivial.

**Value**

The sparsity of the input matrix, as a proportion or a count.

**Author(s)**

Drew Schmidt

**Examples**

```
## Completely sparse matrix
x <- matrix(0, 10, 10)
coop::sparsity(x)

## 15% density / 85% sparsity
x[sample(length(x), size=15)] <- 1
coop::sparsity(x)
```

# Index

\*Topic **package**

coop-package, 2

coop-package, 2

cosine, 3, 4, 5

covar, 4

pcor, 5

sparsity, 3, 6