Package ‘proxyC’

December 10, 2021

Type Package
Title Computes Proximity in Large Sparse Matrices
Version 0.2.4
Description Computes proximity between rows or columns of large matrices efficiently in C++. Functions are optimised for large sparse matrices using the Armadillo and Intel TBB libraries. Among several built-in similarity/distance measures, computation of correlation, cosine similarity and Euclidean distance is particularly fast.

URL https://github.com/koheiw/proxyC
BugReports https://github.com/koheiw/proxyC/issues
License GPL-3
Depends R (>= 3.1.0), methods
Imports Matrix (>= 1.2), Rcpp (>= 0.12.12), RcppParallel
Suggests testthat, entropy, proxy
LinkingTo Rcpp, RcppParallel, RcppArmadillo (>= 0.7.600.1.0)
SystemRequirements C++11
Encoding UTF-8
RoxygenNote 7.1.2
NeedsCompilation yes
Author Kohei Watanabe [cre, aut, cph]
  (<https://orcid.org/0000-0001-6519-5265>),
  Robrecht Cannoodt [aut] (<https://orcid.org/0000-0003-3641-729X>)
Maintainer Kohei Watanabe <watanabe.kohei@gmail.com>
Repository CRAN
Date/Publication 2021-12-10 10:50:02 UTC

R topics documented:

colSds ......................................................... 2
colZeros ...................................................... 2
simil ......................................................... 3
Index

---

**colSds**  
*Standard deviation of columns and rows of large matrices*

**Description**

Produces the same result as `apply(x, 1, sd)` or `apply(x, 2, sd)` without coercing matrix to dense matrix. Values are not identical to `sd` because of the floating point precision issue in C++.

**Usage**

```r
colSds(x)
```

**Arguments**

- `x`: matrix or Matrix object

**Examples**

```r
mt <- Matrix::rsparsematrix(100, 100, 0.01)
colSds(mt)  # the same
```

---

**colZeros**  
*Count number of zeros in columns and rows of large matrices*

**Description**

Produces the same result as applying `sum(x == 0)` to each row or column.

**Usage**

```r
colZeros(x)
```

**Arguments**

- `x`: matrix or Matrix object

**Examples**

```r
mt <- Matrix::rsparsematrix(100, 100, 0.01)
colZeros(mt)  # the same
```
simil

Compute similarity/distance between rows or columns of large matrices

**Description**

Fast similarity/distance computation function for large sparse matrices. You can floor small similarity value to save computation time and storage space by an arbitrary threshold (min_simil) or rank (rank). Please increase the number of threads for better performance using `setThreadOptions`.

**Usage**

```
simil(
  x,
  y = NULL,
  margin = 1,
  method = c("cosine", "correlation", "jaccard", "ejaccard", "dice", "edice", "hamann",
            "simple matching", "faith"),
  min_simil = NULL,
  rank = NULL,
  drop0 = FALSE,
  diag = FALSE,
  use_nan = FALSE,
  digits = 14
)

dist(
  x,
  y = NULL,
  margin = 1,
  method = c("euclidean", "chisquared", "kullback", "manhattan", "maximum", "canberra",
             "minkowski", "hamming"),
  p = 2,
  smooth = 0,
  drop0 = FALSE,
  diag = FALSE,
  use_nan = FALSE,
  digits = 14
)
```

**Arguments**

- `x` matrix or `Matrix` object. Dense matrices are covered to the `CsparseMatrix-class` internally.
- `y` if a `matrix` or `Matrix` object is provided, proximity between documents or features in `x` and `y` is computed.
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>margin</td>
<td>integer indicating margin of similarity/distance computation. 1 indicates rows or 2 indicates columns.</td>
</tr>
<tr>
<td>method</td>
<td>method to compute similarity or distance</td>
</tr>
<tr>
<td>min_simil</td>
<td>the minimum similarity value to be recorded.</td>
</tr>
<tr>
<td>rank</td>
<td>an integer value specifying top-n most similarity values to be recorded.</td>
</tr>
<tr>
<td>drop0</td>
<td>if TRUE, zero values are removed regardless of min_simil or rank.</td>
</tr>
<tr>
<td>diag</td>
<td>if TRUE, only compute diagonal elements of the similarity/distance matrix; useful when comparing corresponding rows or columns of 'x' and 'y'.</td>
</tr>
<tr>
<td>use_nan</td>
<td>if TRUE, return ‘NaN’ if the standard deviation of a vector is zero when ‘method’ is &quot;correlation&quot;; if all the values are zero in a vector when ‘method’ is &quot;cosine&quot;, &quot;kullback&quot; or &quot;chisquared&quot;. Note that use of ‘NaN’ makes the similarity/distance matrix denser and therefore larger.</td>
</tr>
<tr>
<td>digits</td>
<td>determines rounding of small values towards zero. Use primarily to correct rounding errors in C++. See zapsmall.</td>
</tr>
<tr>
<td>p</td>
<td>weight for Minkowski distance</td>
</tr>
<tr>
<td>smooth</td>
<td>adds a fixed value to all the cells to avoid division by zero. Only used when ‘method’ is &quot;chisquared&quot; or &quot;kullback&quot;.</td>
</tr>
</tbody>
</table>

**See Also**

zapsmall

**Examples**

```r
mt <- Matrix::rsparsematrix(100, 100, 0.01)
simil(mt, method = "cosine")[1:5, 1:5]
mt <- Matrix::rsparsematrix(100, 100, 0.01)
dist(mt, method = "euclidean")[1:5, 1:5]
```
Index

colSds, 2
colZeros, 2
CsparseMatrix-class, 3
dist (simil), 3
Matrix, 2, 3
matrix, 2, 3
rowSds (colSds), 2
rowZeros (colZeros), 2
setThreadOptions, 3
simil, 3
zapsmall, 4