Package 'rdist'

October 14, 2022

Title Calculate Pairwise Distances

Version 0.0.5

Description A common framework for calculating distance matrices.

Depends R (>= 3.2.2)

License GPL

URL https://github.com/blasern/rdist

BugReports https://github.com/blasern/rdist/issues

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, methods

RoxygenNote 7.1.0

Suggests testthat

NeedsCompilation yes

Author Nello Blaser [aut, cre]

Maintainer Nello Blaser <nello.blaser@uib.no>

Repository CRAN

Date/Publication 2020-05-04 16:00:02 UTC

R topics documented:

`farthest_point_sampling` *Farthest point sampling*

Description

Farthest point sampling returns a reordering of the metric space P = p_1, ..., p_k, such that each p_i is the farthest point from the first i-1 points.

Usage

```
farthest_point_sampling(
  mat,
  metric = "precomputed",
  k = nrow(mat),
  initial_point_index = 1L,
  return_clusters = FALSE
)
```

Arguments

`mat`	Original distance matrix
`metric`	Distance metric to use (either "precomputed" or a metric from `rdist`)
`k`	Number of points to sample
`initial_point_index`	Index of p_1
`return_clusters`	Should the indices of the closest farthest points be returned?

Examples

```
# generate data
df <- matrix(runif(200), ncol = 2)
dist_mat <- pdist(df)
# farthest point sampling
fps <- farthest_point_sampling(dist_mat)
fps2 <- farthest_point_sampling(df, metric = "euclidean")
all.equal(fps, fps2)
# have a look at the fps distance matrix
rdist(df[fps[1:5], ])
dist_mat[fps, fps][1:5, 1:5]
```

is_metric *Metric and triangle inequality*

Description

Does the distance matric come from a metric

Usage

```
is_distance_matrix(mat, tolerance = .Machine$double.eps^0.5)

triangle_inequality(mat, tolerance = .Machine$double.eps^0.5)
```

Arguments

mat	The matrix to evaluate
tolerance	Differences smaller than tolerance are not reported.

Examples

```
data <- matrix(rnorm(20), ncol = 2)
dm <- pdist(data)
is_distance_matrix(dm)
triangle_inequality(dm)

dm[1, 2] <- 1.1 * dm[1, 2]
is_distance_matrix(dm)
```

product_metric *Product metric*

Description

Returns the p-product metric of two metric spaces. Works for output of ‘rdist‘, ‘pdist‘ or ‘cdist‘.

Usage

```
product_metric(..., p = 2)
```

Arguments

...	Distance matrices or dist objects
p	The power of the Minkowski distance

Examples

```
# generate data
df <- matrix(runif(200), ncol = 2)
# distance matrices
dist_mat <- pdist(df)
dist_1 <- pdist(df[, 1])
dist_2 <- pdist(df[, 2])
# product distance matrix
dist_prod <- product_metric(dist_1, dist_2)
# check equality
all.equal(dist_mat, dist_prod)
```

`rdist` *rdist: an R package for distances*

Description

`rdist` provide a common framework to calculate distances. There are three main functions:

- `rdist` computes the pairwise distances between observations in one matrix and returns a `dist` object,
- `pdist` computes the pairwise distances between observations in one matrix and returns a `matrix`, and
- `cdist` computes the distances between observations in two matrices and returns a `matrix`.

In particular the `cdist` function is often missing in other distance functions. All calculations involving `NA` values will consistently return `NA`.

Usage

```
rdist(X, metric = "euclidean", p = 2L)

pdist(X, metric = "euclidean", p = 2)

cdist(X, Y, metric = "euclidean", p = 2)
```

Arguments

`X, Y`	A matrix
`metric`	The distance metric to use
`p`	The power of the Minkowski distance

Details

Available distance measures are (written for two vectors v and w):

- "euclidean": $\sqrt{\sum_i (v_i - w_i)^2}$
- "minkowski": $(\sum_i |v_i - w_i|^p)^{1/p}$
- "manhattan": $\sum_i (|v_i - w_i|)$
- "maximum" or "chebyshev": $\max_i (|v_i - w_i|)$
- "canberra": $\sum_i (\frac{|v_i - w_i|}{|v_i| + |w_i|})$
- "angular": $\cos^{-1}(cor(v, w))$
- "correlation": $\sqrt{\frac{1 - cor(v,w)}{2}}$
- "absolute_correlation": $\sqrt{1 - |cor(v, w)|^2}$
- "hamming": $(\sum_i v_i \neq w_i) / \sum_i 1$
- "jaccard": $(\sum_i v_i \neq w_i) / \sum_i 1_{v_i \neq 0 \cup w_i \neq 0}$
- Any function that defines a distance between two vectors.

Index