Package 'dissUtils'

August 29, 2016

Type Package

Title Utilities for making pairwise comparisons of multivariate data

Version 1.0

Date 2012-12-06

Author Benjamin N. Taft

Maintainer Benjamin N. Taft <ben.taft@landmarkacoustics.com>

Depends R (>= 2.14)

Suggests MASS (>= 1.0)

SuggestsNote the examples use mvrnorm() from MASS

Description This package has extensible C++ code for computing dissimilarities between vectors. It also has a number of C++ functions for assembling collections of dissimilarities. In particular, it lets you find a matrix of dissimilarities between the rows of two input matrices. There are also functions for finding the nearest neighbors of each row of a matrix, either within the matrix itself or within another matrix.

License GPL (>= 2) | file LICENSE

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-06-02 14:50:54

R topics documented:

dissUtils-package 2
diss 2
diss.index 3
groupwise.density 4
neighbor.density 5
neighbors 6
neighbors.identify 6
unit.hypersphere.volume 7

Index **8**

dissUtils-package *Utilities for making pairwise comparisons of multivariate data*

Description

This package has extensible C++ code for computing dissimilarities between vectors. It also has a number of C++ functions for assembling collections of dissimilarities. In particular, it lets you find a matrix of dissimilarities between the rows of two input matrices. There are also functions for finding the nearest neighbors of each row of a matrix, either within the matrix itself or within another matrix.

Details

Package:	dissUtils
Type:	Package
Version:	0.1
Date:	2012-12-06
License:	GPL (>= 2)

diss	Dissimilarities Between Vectors
diss.index	Convert Indices from Distance Object to Matrix
groupwise.density	Compare Spatial Densities Between Groups
neighbors.identify	Find Neighbor Indices
neighbor.density	N-Dimensional Neighbor Density
neighbors	Find Nearest Neighbor Distances
unit.hypersphere.volume	Helps When Calculating Densities

diss *Many Different Ways to Quantify Dissimilarities Among Multivariate Data*

Description

this function will create a distance object corresponding to the dissimilarities between rows in a matrix `X`, or a matrix of dissimilarities between the rows of matrices `X` and `Y`

Usage

```
diss(X, Y = NULL, method = "euclidean", init.info = NULL)
```

Arguments

X	a matrix of numeric data
Y	a second matrix of numeric data, which must have the same number of columns as X
method	a character string that uniquely matches one of the following:

braycurtis	Bray-Curtis difference, should use proportions
canberra	Canberra difference, should use proportions
chebyshev	Largest difference in any one dimension, like in chess
covariance	You may want to transpose the data before using this
euclidean	multivariate 2-norm
equality	the sum of exactly equal elements in each row
hellinger	Hellinger difference
jaccard	Jaccard distance
mahalanobis	Euclidean distance after scaling and removing covariance, which you can supply with `init.info`
manhattan	The sum of each dimension, no diagonal movement allowed
minkowski	arbitrary n-norm, so that `init.info=2` yields "euclidean" and `init.info = Inf` yields "chebyshev" (but don'
pearson	Pearson product-moment correlation, you may want to transpose the data
procrustes	Doesn't scale or rotate, just treats the vectors as matrices with `init.info` columns and calculates total distance

init.info	some methods require additional information. see above

Value

if `is.null(Y)`, returns a distance object containing pairwise dissimilarities between the points in X.

if `is.matrix(Y)`, returns a `nrow(X)` by `nrow(Y)` matrix containing pairwise dissimilarities between each point in X and each point in Y.

---

diss.index *Convert Indices from Distance Object to Matrix*

Description

Given an index into a distance object of Size N, finds the coordinates of the same pairwise dissimilarity in an N by N matrix of dissimilarities

Usage

```
diss.index(index, N)
```

Arguments

index	the position of the item in the distance object
N	the Size of the distance object, the number of points it compares

See Also

dist

Examples

```
## The function is currently defined as
function (index, N)
{
  i <- floor(.raw.i(n, ix));

  return(c(i = i, j = .calc.j(i, n, ix)));
}
```

---

groupwise.density *Searches Subsets for Nearest Neighbor Densities*

Description

In order to compare the distributions of different groups within the same multivariate space, calculates the nearest-neighbor densities of each point in the whole data set according to the distribution of each subset.

Usage

```
groupwise.density(X, groups, method = "euclidean", p.neighbors = 0.01, init.info = NULL)
```

Arguments

X	a matrix of numeric data
groups	a factor or vector that can be coerced into a factor, specifying which group each row of X belongs to.
method	see diss
p.neighbors	the proportion of each groups neighbors that should be visited. Proportions are necessary when groups have different sizes because otherwise the densities aren't as comparable.
init.info	see diss

Value

an nrow(X) by nlevels(as.factor(groups)) matrix of nearest-neighbor density estimates.

neighbor.density *Calculate Multidimensional Densities from Neighbor Distances*

Description

neighbor.density estimates the density around a point by accounting for the dimensionality of the space the neighbors are in, the total number of points in the space, and how many neighbors are found at least as close to the point as the density given.

Usage

```
neighbor.density(neigh.dists, D, k, N)
```

Arguments

neigh.dists	a vector of distances between members of a multivariate data set and their kth-nearest neighbor
D	the number of dimensions of the multivariate space
k	the number of neighbors found around each point within the hyperspheres with radii given in neigh.dists
N	the total number of points in the data set from which the neighbors are drawn. This may not be equal to length(neigh.dists) if the neighbors are in a separate data set from the points of interest.

Value

a numeric vector of densities

References

http://en.wikipedia.org/wiki/N-sphere

Examples

```
## The function is currently defined as
function (neigh.dists, D, k, N)
{
    radius <- unit.hypersphere.volume(D)
    return(k/(N * radius * neigh.dists))
  }
```

neighbors *Find Nearest Neighbor Distances*

Description

Given one (or two) multivariate data sets, a difference method, and k neighbors to search for, `neighbors` finds the k points in the data set (or the second data set) that are closest to each point in the data set (or the first data set)

Usage

```
neighbors(X, Y = NULL, method = "euclidean", n.neighbors = 1, init.info = NULL)
```

Arguments

`X`	a matrix of numeric values
`Y`	an optional second matrix that must have the same number of columns as `X`
`method`	one of the method choices from `diss`
`n.neighbors`	an integer between 1 and nrow(X) (or nrow(Y), if it is not null)
`init.info`	some difference methods require additional information. see `diss`

Value

returns an `nrow(X)` by `n.neighbors` matrix of distances

neighbors.identify *Find Neighbor Indices*

Description

Uses a distance object and a vector of known distances to identify the neighbors that correspond to those distances.

Usage

```
neighbors.identify(neighbor.matrix, all.dists)
```

Arguments

`neighbor.matrix`	a matrix of distances to neighbors
`all.dists`	either a distance object or a matrix of distances such as is produced by `diss`

Value

a `dim(neighbor.matrix)` matrix of integer indices between 1 and `all.dists$Size` or `ncol(all.dists)`

See Also

`diss`, `dist`

`unit.hypersphere.volume`

Helps When Calculating Densities

Description

Finds the volume of a hypersphere in R^D with radius one.

Usage

```
unit.hypersphere.volume(D)
```

Arguments

`D`	the number of dimensions that the hypersphere extends into

Value

the volume of the unit hypersphere

References

http://en.wikipedia.org/wiki/N-sphere

Index

∗Topic **multivariate**
diss, 2
diss.index, 3
dissUtils-package, 2
groupwise.density, 4
neighbor.density, 5
neighbors, 6
neighbors.identify, 6
unit.hypersphere.volume, 7
∗Topic **package**
dissUtils-package, 2

diss, 2, *4*, *6*, *7*
diss.index, 3
dissUtils *(*dissUtils-package*)*, 2
dissUtils-package, 2
dist, *4*, *7*

groupwise.density, 4

neighbor.density, 5
neighbors, 6
neighbors.identify, 6

unit.hypersphere.volume, 7