Package ‘spe’

February 20, 2015

Version 1.1.2
Date 2009-02-24
Title Stochastic Proximity Embedding
Author Rajarshi Guha <rajarshi.guha@gmail.com>
Maintainer Rajarshi Guha <rajarshi.guha@gmail.com>
License GPL
ZipData no
NeedsCompilation yes
Repository CRAN
Date/Publication 2009-02-24 21:29:28

R topics documented:

\begin{itemize}
  \item eval.stress \hfill 1
  \item phone \hfill 3
  \item sample.max.distance \hfill 3
  \item spe \hfill 4
  \item swissroll \hfill 6
\end{itemize}

Index 7

\begin{itemize}
  \item eval.stress \textit{Evaluates the Sammon stress of an embedding}
\end{itemize}

Description

Given an N dimensional dataset embedded M dimensions, this function will evaluate the Sammon stress of the embedding, via probability sampling
Usage

eval.stress( x, coord,
            ndim = 0, edim = 0, nobs = 0,
            samplesize = 1e6)

Arguments

x The embedded data in matrix form. If present in a data.frame it will be coerced to a matrix
coord The input data in matrix form. If present in a data.frame it will be coerced to a matrix
nobs The number of observations (rows of the input matrix should be the same as the rows of the embedding matrix) If it is not specified nobs will be taken as nrow(coord)
ndim The number of input dimensions. If not specified it will be taken as ncol(coord)
edim The number of dimensions to embed in. If not specified it will be taken as ncol(x)
samplesize The number of iterations for probability sampling. For a dataset of 6070 observations there will be 6070x6069/2 pairwise distances. The default value gives a close approximation and runs fast. If you want a better approximation 1e7 is a good value. YMMV

Details

The Sammon stress is given by

\[ S = \sum_{i<j} \frac{(d_{ij} - r_{ij})^2}{r_{ij}} / \sum_{i<j} r_{ij} \]

where \( d_{ij} \) is the Euclidean distance between two observations in the embedded data and \( r_{ij} \) is the relationship (in this case it is the Euclidean distance but could be a similarity value) between the two observations in the input data

Value

Returns the value of the Sammon stress as a single number

References


See Also

spe
**Description**

This dataset consists of the 3D points representing an image of a phone. The original data was stripped of connectivity information and centroids of the triangles were added to generate additional points.

**Usage**

```r
data(phone)
```

**Format**

A data.frame with 6070 rows and 3 columns.

**Source**


---

**sample.max.distance**

*Samples the distances in the input dimensions to get the maximum distance*

**Description**

The maximum distance in the input dimensions is required to generate a value of the neighborhood radius. For datasets with >1000 observations an all pairs calculation is prohibitive. Instead probability sampling is used so that two points are randomly chosen and their distance is calculated. This is repeated for a user specified number of times and the maximum distance obtained is kept track of and returned at the end.

**Usage**

```r
sample.max.distance( coord,
                      nobs = 0, ndim = 0,
                      samplesize = 1e6)
```
Arguments

coord: The input data in matrix form. If present in a data.frame it will be coerced to a matrix.
nobs: The number of observations (rows of the input matrix should be the same as the rows of the embedding matrix). If it is not specified nobs will be taken as nrow(coord).
ndim: The number of input dimensions. If not specified it will be taken as ncol(coord).
samplesize: The number of iterations for probability sampling. For a dataset of 6070 observations there will be 6070×6069/2 pairwise distances. The default value gives a close approximation and runs fast. If you want a better approximation 1e7 is a good value. YMMV.

Value

Returns the value of the maximum distance found as a single number.

See Also

spe

Description

Embeds an N dimensional dataset in M dimensions, such that distances (or similarities) in the original N dimensions are maintained (as close as possible) in the final M dimensions.

Usage

```r
spe( coord, rcutpercent = 1, maxdist = 0,
    nobs = 0, ndim = 0, edim,
    lambda0 = 2.0, lambda1 = 0.01,
    nstep = 1e6, ncycle = 100,
    evalstress=FALSE, sampledist=TRUE, samplesize = 1e6)
```

Arguments

coord: This should be a matrix with number of rows equal to the number of observations and number of columns equal to the input dimension. A data.frame may also be supplied and it will be converted to a matrix (so all names will be lost).
rcutpercent: This is the percentage of the maximum distance (as determined by probability sampling) that will be used as the neighborhood radius. Setting rcutpercent to a value greater than 1 effectively sets it to infinity.
maxdist If you have already calculated a maximum distance then you can supply it and probability sampling will not be carried out to obtain a maximum distance. The default is to carry out sampling. By setting maxdist to a non zero value sampling will not be carried out (even if sampledist=TRUE)

nobs The number of observations. If it is not specified nobs will be taken as nrow(coord)

ndim The number of input dimensions. If not specified it will be taken as ncol(coord)

edim The number of dimensions to embed in

lambda0 The starting value of the learning parameter

lambda1 The ending value of the learning parameter

nstep The number of refinement steps

ncycle The number of cycles to carry out refinement for

evalstress If TRUE the function will evaluate the Sammon stress on the final embedding

sampledist If TRUE an approximation to the maximum distance in the input dimensions will be obtained via probability sampling

samplesize The number of iterations for probability sampling. For a dataset of 6070 observations there will be 6070x6069/2 pairwise distances. The default value gives a close approximation and runs fast. If you want a better approximation 1e7 is a good value. YMMV

Details

Efficient determination of rcut is yet to be implemented (using the connected component method). As a result you will have to determine a value of rcutpercent by trial and error. The pivot SPE method (J. Mol. Graph. Model., 2003, 22, 133-140) is not yet implemented

Value

If evalstress is TRUE it will be a list with two components named x and stress. x is the matrix of the final embedding and stress is the final stress

Author(s)

Rajarshi Guha <rajarshi@presidency.com>

References


See Also

eval.stress, sample.max.distance
Examples

```r
## load the phone dataset
data(phone)

## run SPE, embed$stress should be 0 or very close to it
## You can plot the embedding using the scatterplot3d package
## (This will take a few minutes to run)
embed <- spe(phone, edim=3, evalstress=TRUE)

## evaluate the Sammon stress
stress <- eval.stress(embed$X, phone)

## embed the Swiss Roll dataset in 2D
data(swissroll)
embed <- spe(swissroll, edim=2, evalstress=TRUE)
```

---

**swissroll**

*Data points for the Swiss Roll function in 3 dimensions*

**Description**

This dataset comprise 1000 3D points generated using:

\[
x = \phi \cos \phi, \quad y = \phi \sin \phi, \quad z
\]

**Usage**

`data(swissroll)`

**Format**

A data.frame with 1000 rows and 3 column
Index

*Topic **datasets**
  phone, 3
  swissroll, 6
*Topic **nonparametric**
  eval.stress, 1
  sample.max.distance, 3
  spe, 4

  eval.stress, 1, 5

  phone, 3

  sample.max.distance, 3, 5
  spe, 2, 4, 4
  swissroll, 6