Package ‘mp’

August 15, 2016

Type Package
Title Multidimensional Projection Techniques
Version 0.4.1
Date 2016-08-13
Author Francisco M. Fatore, Samuel G. Fadel
Maintainer Francisco M. Fatore <fmfatore@gmail.com>
Description Multidimensional projection techniques are used to create two dimensional representations of multidimensional data sets.
License GPL
Depends R (>= 1.8.0)
Imports Rcpp (>= 0.11.0)
LinkingTo Rcpp, RcppArmadillo
RoxygenNote 5.0.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2016-08-15 16:26:10

R topics documented:

  forceScheme ........................................... 2
  is.symmetric ......................................... 3
  lamp .................................................. 3
  lsp ..................................................... 4
  mp ..................................................... 5
  pekalska ............................................. 5
  plmp ................................................... 6
  tSNE .................................................. 7

Index 9
forceScheme

Description

Creates a 2D representation of the data based on a dissimilarity matrix. A few modifications have been made in relation to the method described in the literature: shuffled indices are used to minimize the order dependency factor, only a fraction of delta is used for better stability and a tolerance factor was introduced as a second stop criterion.

Usage

forceScheme(D, Y = NULL, max.iter = 50, tol = 0, fraction = 8, eps = 1e-05)

Arguments

- **D**: A dissimilarity structure such as that returned by dist or a full symmetric matrix containing the dissimilarities.
- **Y**: Initial 2D configuration. A random configuration will be used when omitted.
- **max.iter**: Maximum number of iterations that the algorithm will run.
- **tol**: The tolerance for the accumulated error between iterations. If set to 0, the algorithm will run max.iter times.
- **fraction**: Controls the point movement. Larger values mean less freedom to move.
- **eps**: Minimum distance between two points.

Value

The 2D representation of the data.

References


See Also

dist (stats) and dist (proxy) for d computation
is.symmetric

Tests whether the given matrix is symmetric.

is.symmetric(mat)

Arguments

mat Matrix to be tested for symmetry.

Value

Whether the matrix is symmetric.

lamp Local Affine Multidimensional Projection

Description

Creates a 2D representation of the data. Requires a subsample (sample.indices) and its 2D representation (Ys).

Usage

lamp(X, sample.indices = NULL, Ys = NULL, cp = 1)
Arguments

X A data frame or matrix.
sample.indices The indices of data points in X used as subsamples. If not given, some points from X will be randomly selected and Ys will be generated by calling forceScheme on them.
Ys Initial 2D configuration of the data subsamples (will be ignored if sample.indices is NULL). Scaling the columns to [-0.5, 0.5] is recommended to avoid scaling problems.

Value

The 2D representation of the data.

References


Examples

# Iris example
emb <- lamp(iris[, 1:4])
plot(emb, col=iris$Species)

---

lsp Least-Square Projection

Description

Creates a q-dimensional representation of multidimensional data. Requires a subsample (sample.indices) and its qD representation (Ys).

Usage

lsp(X, sample.indices = NULL, Ys = NULL, k = 15, q = 2)

Arguments

X A data frame or matrix.
sample.indices The indices of data points in X used as subsamples. If not given, some rows from X will be randomly selected and Ys will be generated by calling forceScheme on them.
**mp**  

Ys  
Initial kD configuration of the data subsamples (will be ignored if sample.indices is NULL).

k  
Number of neighbors used to build the neighborhood graph.

q  
The target dimensionality.

**Value**

The qD representation of the data.

**References**


**Examples**

```r
# Iris example
emb <- lsp(iris[, 1:4])
plot(emb, col=iris$Species)
```

---

**pekalska**  

*Pekalska’s approach to speeding up Sammon’s mapping.*

**Description**

Creates a k-dimensional representation of the data. As input, a subsample and its k-dimensional mapping are required. The method approximates the subsample mapping to a linear mapping based on the distances matrix of the subsample and then applies the same mapping to all instances.

**Usage**

`pekalska(D, sample.indices = NULL, Ys = NULL)`

**Arguments**

- **D**  
  dist object or distances matrix.
- **sample.indices**  
  The indices of subsamples.
- **Ys**  
  The subsample mapping (k-dimensional).
Value

The low-dimensional representation of the data.

References


---

plmp

**Part-Linear Multidimensional Projection**

Description

Creates a k-dimensional representation of the data. As input, a subsample and its k-dimensional mapping (control points) are required. The method approximates the subsample mapping to a linear mapping and then applies the same mapping to all instances.

Usage

`plmp(x, sample.indices = NULL, Ys = NULL, k = 2)`

Arguments

- `x`: A dataframe or matrix representing the data.
- `sample.indices`: The indices of subsamples used as control points.
- `Ys`: The control points.
- `k`: The target dimensionality.

Value

The low-dimensional representation of the data.

References


Examples

```r
# Iris example
emb <- plmp(iris[,1:4])
plot(emb, col=iris$Species)
```
tSNE

\textit{t-Distributed Stochastic Neighbor Embedding}

\textbf{Description}

Creates a k-dimensional representation of the data by modeling the probability of picking neighbors using a Gaussian for the high-dimensional data and t-Student for the low-dimensional map and then minimizing the KL divergence between them. This implementation uses the same default parameters as defined by the authors.

\textbf{Usage}

\begin{verbatim}
tSNE(X, Y = NULL, k = 2, perplexity = 30, n.iter = 1000, eta = 500, initial.momentum = 0.5, final.momentum = 0.8, early.exaggeration = 4, gain.fraction = 0.2, momentum.threshold.iter = 20, exaggeration.threshold.iter = 100, max.binsearch.tries = 50)
\end{verbatim}

\textbf{Arguments}

- \texttt{X}  
  A data frame, data matrix, dissimilarity (distance) matrix or dist object.
- \texttt{Y}  
  Initial k-dimensional configuration. If NULL, the method uses a random initial configuration.
- \texttt{k}  
  Target dimensionality. Avoid anything other than 2 or 3.
- \texttt{perplexity}  
  A rough upper bound on the neighborhood size.
- \texttt{n.iter}  
  Number of iterations to perform.
- \texttt{eta}  
  The "learning rate" for the cost function minimization
- \texttt{initial.momentum}  
  The initial momentum used before changing
- \texttt{final.momentum}  
  The momentum to use on remaining iterations
- \texttt{early.exaggeration}  
  The early exaggeration applied to initial iterations
- \texttt{gain.fraction}  
  Undocumented
- \texttt{momentum.threshold.iter}  
  Number of iterations before using the final momentum
- \texttt{exaggeration.threshold.iter}  
  Number of iterations before using the real probabilities
- \texttt{max.binsearch.tries}  
  Maximum number of trials in binary search for parameters to achieve the target perplexity

\textbf{Value}

The k-dimensional representation of the data.
References


Examples

# Iris example
emb <- tsNE(iris[, 1:4])
plot(emb, col=iris$Species)
Index

dist, 2
forceScheme, 2
is.symmetric, 3
lamp, 3
lsp, 4

mp, 5
mp-package (mp), 5

pekalska, 5
plmp, 6

tSNE, 7