Package ‘hydra’

October 13, 2022

Type Package
Title Hyperbolic Embedding
Version 0.1.0
Author Martin Keller-Ressel
Maintainer Martin Keller-Ressel <martin.keller-ressel@tu-dresden.de>
Description Calculate an optimal embedding of a set of data points into low-dimensional hyperbolic space. This uses the strain-minimizing hyperbolic embedding of Keller-Ressel and Nargang (2019), see <arXiv:1903.08977>.
Suggests igraph, igraphdata, Matrix, RSpectra
License GPL-2
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
NeedsCompilation no
Repository CRAN
Date/Publication 2019-04-04 16:10:07 UTC

R topics documented:

  hydra ................................................................. 2
  hydraPlus ......................................................... 4
  karate ............................................................. 6
  plot.hydra ....................................................... 6

Index 8
**Description**

Implements the HYDRA (hyperbolic distance recovery and approximation) method for embedding high-dimensional data points (represented by their distance matrix \( D \)) into low-dimensional hyperbolic space.

**Usage**

```r
hydra(D, dim = 2, curvature = 1, alpha = 1.1, equi.adj = 0.5, 
control = list())
```

**Arguments**

- **D**
  - a square symmetric matrix of distances (or dissimilarities) to be embedded, can also be a `dist` object

- **dim**
  - embedding dimension

- **curvature**
  - embedding curvature; if this argument is NULL, hydra tries to find the optimal curvature

- **alpha**
  - real number greater one; adjusts the hyperbolic curvature. Values larger than one yield a more distorted embedding where points are pushed to the outer boundary (i.e. the ideal points) of hyperbolic space. The interaction between curvature and alpha is non-linear.

- **equi.adj**
  - equi-angular adjustment; must be a real number between zero and one; only used if \( \text{dim} = 2 \). Value 0 means no adjustment, 1 adjusts embedded data points such that their angular coordinates in the Poincare disc are uniformly distributed. Other values interpolate between the two extremes. Setting the parameter to non-zero values can make the embedding result look more harmonious in plots.

- **control**
  - a list which may contain the following boolean flags:
    - **polar** - return polar coordinates in dimension 2 (default: TRUE if \( \text{dim} = 2 \). This flag is ignored in higher dimension)
    - **isotropic.adj** - perform isotropic adjustment, ignoring Eigenvalues (default: TRUE if \( \text{dim} = 2 \), FALSE else)
    - **return.lorentz** - return raw Lorentz coordinates (before projection to hyperbolic space) (default: FALSE)
    - **return.stress** - return embedding stress (default: TRUE)
    - **return.dist** - return hyperbolic distance matrix of embedded points (default: FALSE)
    - **use.eigs** - use `eigs` function from `RSpectra` and `norm` function from `Matrix` to speed up computation (default: FALSE)
**Details**


**Value**

A `‘hydra’` object, which is a list with all or some of the following components:

- **r** a vector containing the radial coordinates of the embedded points
- **directional** a matrix with `dim` columns containing as rows the directional coordinates of the embedded points
- **theta** a vector containing the angular coordinates of the embedded points (only returned if `dim` is 2 and `polar` flag is TRUE)
- **curvature** the curvature used for the returned embedding
- **dim** the dimension used for the returned embedding
- **stress** the stress (i.e. the mean-square difference) between distances supplied in `D` and the hyperbolic distance matrix of the returned embedding
- **dist** the hyperbolic distance matrix of the returned embedding (only returned if flag `return.dist` is true. Computation may be time- and memory-intensive.)
- **x0** a vector containing the 'time-like' coordinate of the raw Lorentz embedding (only returned if flag `return.lorentz` is true)
- **X** a matrix with `dim` columns containing as rows the 'space-like' coordinate of the raw Lorentz embedding (only returned if flag `return.lorentz` is true)

**Author(s)**

Martin Keller-Ressel <martin.keller-ressel@tu-dresden.de>

**Examples**

```r
data(karate)
embedding <- hydra(karate$distance)
plot(embedding,labels=karate$label,lab.col=karate$group,graph.adj=karate$adjacency)

## Compare with Multidimensional scaling (MDS):
mds <- cmdscale(karate$distance) # Compute Euclidean embedding with MDS
mds.stress <- sqrt(sum((as.matrix(dist(mds)) - karate$distance)^2)) # Calculate embedding stress
cc(.embedding$stress, mds.stress) # Compare hyperbolic with Euclidean stress
```
hydraPlus

hydra with additional stress minimization

Description

Runs the hydra method and then performs a further optimization step by minimizing the stress of the embedding and optimizing hyperbolic curvature.

Usage

hydraPlus(D, dim = 2, curvature = 1, alpha = 1.1, equi.adj = 0.5, control = list(), curvature.bias = 1, curvature.freeze = TRUE, curvature.max = NULL, maxit = 1000, ...)

Arguments

D a square symmetric matrix of distances (or dissimilarities) to be embedded, can also be a dist object

dim embedding dimension

curvature embedding curvature; if this argument is NULL, hydra tries to find the optimal curvature

alpha real number greater one; adjusts the hyperbolic curvature. Values larger than one yield a more distorted embedding where points are pushed to the outer boundary (i.e. the ideal points) of hyperbolic space. The interaction between curvature and alpha is non-linear.

equi.adj equi-angular adjustment; must be a real number between zero and one; only used if dim is 2. Value 0 means no adjustment, 1 adjusts embedded data points such that their angular coordinates in the Poincare disc are uniformly distributed. Other values interpolate between the two extremes. Setting the parameter to non-zero values can make the embedding result look more harmoniuous in plots.

control a list which may contain the following boolean flags:

• polar - return polar coordinates in dimension 2 (default: TRUE if dim is 2. This flag is ignored in higher dimension)

• isotropic.adj - perform isotropic adjustment, ignoring Eigenvalues (default: TRUE if dim is 2, FALSE else)

• return.lorentz - return raw Lorentz coordinates (before projection to hyperbolic space) (default: FALSE)

• return.stress - return embedding stress (default: TRUE)

• return.dist - return hyperbolic distance matrix of embedded points (default: FALSE)

• use.eigs - use eigs function from RSpectra and norm function from Matrix to speed up computation (default: FALSE)

curvature.bias Modify curvature before stress minimization by multiplying with curvature.bias
`curvature.freeze`  
Freeze the curvature returned by `hydra`. If TRUE then no optimization of curvature is attempted in the second stage of the algorithm. If FALSE then curvature is optimized in the second stage.

`curvature.max`  
Upper bound for the curvature. If NULL, a default bound is used.

`maxit`  
Maximal number of iterations. This parameter is passed to the optimization routine `optim`.

Additional parameters are passed to `optim`, which performs the underlying stress minimization.

Details  

Value  
A 'hydra' object, which is a list with all or some of the following components:

- `r` a vector containing the radial coordinates of the embedded points
- `directional` a matrix with `dim` columns containing as rows the directional coordinates of the embedded points
- `theta` a vector containing the angular coordinates of the embedded points (only returned if `dim` is 2 and `polar` flag is TRUE)
- `curvature` the curvature used for the returned embedding
- `dim` the dimension used for the returned embedding
- `stress` the stress (i.e. the mean-square difference) between distances supplied in `D` and the hyperbolic distance matrix of the returned embedding
- `dist` the hyperbolic distance matrix of the returned embedding (only returned if flag `return.dist` is true. Computation may be time- and memory-intensive.)
- `x0` a vector containing the 'time-like' coordinate of the raw Lorentz embedding (only returned if flag `return.lorentz` is true)
- `X` a matrix with `dim` columns containing as rows the 'space-like' coordinate of the raw Lorentz embedding (only returned if flag `return.lorentz` is true)

Author(s)  
Martin Keller-Ressel <martin.keller-ressel@tu-dresden.de>

Examples  
```r
data(karate)
embedding <- hydraPlus(karate$distance)
plot(embedding, labels=karate$label, node.col=karate$group, graph.adj=karate$adjacency)```
karate  

Zachary's karate club network data

Description

The social network of a university karate club described in the paper "An Information Flow Model for Conflict and Fission in Small Groups" by Wayne W. Zachary.

Edges indicate social interactions between members

Usage

data(karate)

Format

A list with the elements:

- adjacency  The adjacency matrix of the karate club network
- distances  The shortest-path distance between nodes
- labels  Node labels. 'A' and 'H' are John A. and Mr. Hi, who led the two groups after the split of the club
- group  The two groups after the split

plot.hydra  

Plot a hyperbolic embedding

Description

Plot a two-dimensional hyperbolic embedding as returned by hydra in the Poincare disc

Usage

```r
## S3 method for class 'hydra'
plot(x, labels = NULL, node.col = 1, pch = NULL,
     graph.adj = NULL, crop.disc = TRUE, shrink.disc = FALSE,
     disc.col = "grey90", rotation = 0, mark.center = 0,
     mark.angles = 0, mildify = 3, cex = 1, ...)
```
Arguments

- **x**: a hydra object as returned by `hydra` with dimension `dim` equal 2
- **labels**: character labels for the embedded points, supplied as a vector. NULL triggers default values
- **node.col**: colors for the labels and/or points, supplied as a vector. NULL triggers default values. See ‘Color Specification’ in `par` for details
- **pch**: plotting ‘characters’ for the embedded points. supplied as a vector. NULL triggers default values. See `points` for details
- **graph.adj**: a graph adjacency matrix that is used to plot links between the embedded points (links are drawn for all non-zero elements of `graph.adj`)
- **crop.disc**: should the Poincare disc be cropped or fully shown? Defaults to TRUE
- **shrink.disc**: if true, the Poincare disc is shrunk to tightly fit all plotted points. Defaults to FALSE
- **disc.col**: color of the Poincare disc. Set to "white" to hide disc
- **rotation**: rotate points by this angle (specified in degrees) around the center of the Poincare disc
- **mark.center**: Should a cross be placed at the center of the disc? If 0, nothing is drawn. Other values specify the relative size of the cross mark.
- **mark.angles**: Should the angular coordinates of points be marked at the boundary of the disc? If 0, nothing is drawn. Other values specify the relative size of the angle marks.
- **mildify**: large values reduce the curvature of links. Values around 3 are visually most appealing. Setting `mildify` to 1 shows the true hyperbolic curvature
- **cex**: character expansion for labels and points, supplied as a numerical vector. See also `points`

... all other parameters are passed on as additional graphical parameters (see `par`)

Author(s)

Martin Keller-Ressel <martin.keller-ressel@tu-dresden.de>

Examples

data(karate)
embedding <- hydra(karate$distance)
plot(embedding, labels=karate$label, node.col=karate$group, graph.adj=karate$adjacency)

# plot points instead of labels, hide Poincare disc and rotate by 90 degrees:
plot(embedding, pch=karate$group, node.col=karate$group, graph.adj=karate$adjacency, disc.col="white", rotation=90)

# do not crop the Poincare disc, mark the center and mark angles:
plot(embedding, labels=karate$label, node.col=karate$group, graph.adj=karate$adjacency, crop.disc=FALSE, mark.center=0.05, mark.angles=0.025)
Index

dist, 2, 4

eigs, 2, 4

hydra, 2, 4–7
hydraPlus, 4

karate, 6

norm, 2, 4

optim, 5

par, 7

plot.hydra, 6
points, 7