Package ‘liminal’

May 28, 2021

Type Package

Title Multivariate Data Visualization with Tours and Embeddings

Version 0.1.2

Description Compose interactive visualisations designed for exploratory high-dimensional data analysis. With 'liminal' you can create linked interactive graphics to diagnose the quality of a dimension reduction technique and explore the global structure of a dataset with a tour. A complete description of the method is discussed in ['Lee' & 'Laa' & 'Cook' (2020) <arXiv:2012.06077>].

License MIT + file LICENSE

Encoding UTF-8

LazyData true


Depends R (>= 3.6.0)

Imports tourr (>= 0.6.0), shiny, vegawidget, miniUI, jsonlite, rlang, dplyr, matrixStats

RoxygenNote 7.1.1

Suggests Rtsne, knitr, rmarkdown, covr, ggplot2, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Stuart Lee [aut, cre, cph] (<https://orcid.org/0000-0003-1179-8436>)

Maintainer Stuart Lee <stuart.andrew.lee@gmail.com>

Repository CRAN

Date/Publication 2021-05-28 07:30:05 UTC
clamp

R topics documented:

clamp ................................................. 2
compute_half_range ................................. 3
compute_proj_dist ................................. 4
fake_trees ........................................ 4
limn_pal_tableau10 .................................. 5
limn_tour ........................................... 6
limn_tour_link ..................................... 7
morph_center ...................................... 9
pdfsense ......................................... 10

Index 12

clamp  Rescale all columns of a matrix

Description
Rescale all columns of a matrix

Usage
clamp(.data)

clamp_robust(.data)

clamp_sd(.data, sd = 1)

clamp_standardize(.data, sd = 1)

Arguments
.data  A numeric matrix
sd  the value of each columns standard deviation (default is 1)

Details
These functions are used internally by the tour to rescale all columns of .data.

- clamp() rescales so all values for each column lie in the unit interval
- clamp_robust() rescales by first centering by the median and then scaling by the median absolute deviation.
- clamp_sd() rescales all columns to have a fixed standard deviation.
- clamp_standardize() rescales all columns to have zero mean and unit variance.
**compute_half_range**

**Value**
A matrix with the same dimension as `.data` where each column has been rescaled.

**Examples**
```
mv <- matrix(rnorm(30), ncol = 3)
clamp(mv)
clamp_robust(mv)
clamp_sd(mv)
clamp_standardize(mv)
```

**compute_half_range**  
Compute range of axes for a tour

**Description**
Compute range of axes for a tour

**Usage**
```
compute_half_range(.data, center = TRUE)
```

**Arguments**
- `.data` A numeric matrix
- `center` Subtract `colMeans(.data)` from each column in `.data`? Default is TRUE.

**Details**
This function computes the maximum squared Euclidean distance of rows in a matrix like object. Mostly used internally for setting up xy-axis ranges for a tour animation.

**Value**
A numeric vector of length 1.

**Examples**
```
mv <- matrix(rnorm(300), ncol = 3)
compute_half_range(mv)
compute_half_range(mv, center = FALSE)
```
compute_proj_dist  
Compute Frobenius norm of matrix-like objects x and y

Description
Compute Frobenius norm of matrix-like objects x and y

Usage
compute_proj_dist(x, y)

Arguments
x, y  'matrix' like objects that have tcrossprod methods

Value
A numeric vector of length 1 that is the Frobenius norm

Examples
x <- matrix(rnorm(300), ncol = 3)
y <- matrix(rnorm(300), ncol = 3)
compute_proj_dist(x, y)

fake_trees  
A high-dimensional tree data structure with 10 branching points.

Description
A high-dimensional tree data structure with 10 branching points.

Usage
fake_trees

Format
An object of class data.frame with 3000 rows and 101 columns.

Details
Data are obtained from diffusion limited aggregation tree simulation in the phate python and phateR packages, but reconstructed as a wide data.frame rather than a list.
There are 3000 rows and 101 columns, the first 100 columns are labelled dim1 - dim100, and are numeric, while the final column is a factor representing the branch id.
**Source**

PHATE

---

**limn_pal_tableau10**  liminal color palettes

---

**Description**

liminal color palettes

**Usage**

```r
limn_pal_tableau10()
limn_pal_tableau20()
```

**Details**

Vectors of colors based on the schemes available in Vega-Lite. Their main purpose is so you can use these palettes in ggplot2 graphics, so that graphs align with the `limn_tour()` functions.

**Value**

A character vector of hex color codes of length 10 or 20.

**See Also**

https://vega.github.io/vega/docs/schemes/

**Examples**

```r
if (requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)
  ggplot(fake_trees, aes(x = dim1, y = dim2, color = branches)) +
    geom_point() +
    scale_color_manual(values = limn_pal_tableau10())

  ggplot(fake_trees, aes(x = dim1, y = dim2, color = branches)) +
    geom_point() +
    scale_color_manual(values = limn_pal_tableau20())
}
```
limn_tour  

*Tour a high dimensional dataset*

**Description**

Tour a high dimensional dataset

**Usage**

```r
limn_tour(
  tour_data,
  cols,
  color = NULL,
  tour_path = tourr::grand_tour(),
  rescale = clamp,
  morph = "center",
  gadget_mode = TRUE
)
```

**Arguments**

- `tour_data` : a data.frame to tour
- `cols` : Columns to tour. This can use a tidyselect specification such as `tidyselect::starts_with()`.
- `color` : A variable mapping to the color aesthetic, if NULL points will be colored black.
- `tour_path` : the tour path to take, the default is `tourr::grand_tour()` but also works with `tourr::guided_tour()`.
- `rescale` : A function that rescales `cols`, the default is to `clamp()` the data to lie in the hyperdimensional unit cube. To not perform any scaling use `identity()`.
- `morph` : One of `c("center", "centre", "identity", "radial")` that rescales each projection along the tour path. The default is to center the projections and divide by half range. See `morph_center()` for details.
- `gadget_mode` : Run the app as a `shiny::runGadget()` which will load the app in the RStudio Viewer pane or a browser (default = TRUE). If FALSE will return a regular shiny app object that could be used to deploy the app elsewhere.

**Details**

The tour interface consists of two views:

1. the tour view which is a dynamic scatterplot
2. the axis view which shows the direction and magnitude of the basis vectors being generated.

There are several other user controls available:

- A play button, that when pressed will start the tour animation.
- A pause button, that when pressed will pause the tour animation.
• The title of the view includes the half range. The half range is a scale factor for projections and can be thought of as a way of zooming in and out on points. It can be modified by scrolling (via a mouse-wheel movement). Double-click to reset to the default tour view.

• If categorical variable has been used, the legend can be toggled to highlight categories of interest with shift + mouse click. Multiple categories can be selected in this way. To reset double click the legend title.

• Brushing is activated by moving the mouse on the tour view. If the tour animation a brush event will pause it.

Value

The tour interface loads a shiny app either in the Viewer pane if you are using Rstudio or in a browser window. After iterating through the tour and and highlighting subsets of interest, you can click the ‘Done’ button. This will return a named list with two elements:

• selected_basis: a matrix consisting of the final projection selected
• tour_brush_box: a list consisting of the bounding box of brush
• tour_half_range: the current value of half range parameter

See Also

compute_half_range(), morph_center(), limn_tour_link()

Examples

if (interactive()) {
  # tour the first ten columns of the fake tree data
  # loads the default interface
  limn_tour(fake_trees, dim1:dim10)
  # perform the same action but now coloring points
  limn_tour(fake_trees, dim1:dim10, color = branches)
}

limn_tour_link

Link a 2-d embedding with a tour

Description

Link a 2-d embedding with a tour

Usage

limn_tour_link(
  embed_data,
  tour_data,
  cols = NULL,
  color = NULL,
tour_path = tourr::grand_tour(),
rescale = clamp,
morph = "center",
gadget_mode = TRUE
)

Arguments

embed_data  A data.frame representing embedding coordinates
tour_data   a data.frame to tour
cols       Columns to tour. This can use a tidyselect specification such as tidyselect::starts_with().
color      A variable mapping to the color aesthetic, if NULL points will be colored black.
tour_path  the tour path to take, the default is tourr::grand_tour() but also works with tourr::guided_tour().
rescale    A function that rescales cols, the default is to clamp() the data to lie in the hyperdimensional unit cube. To not perform any scaling use identity().
morph      One of c("center","centre","identity","radial") that rescales each projection along the tour path. The default is to center the projections and divide by half range. See morph_center() for details.
gadget_mode Run the app as a shiny::runGadget() which will load the app in the RStudio Viewer pane or a browser (default = TRUE). If FALSE will return a regular shiny app object that could be used to deploy the app elsewhere.

Details

All controls for the app can be obtained by clicking on the help button, in the bottom panel. More details are described below:

• The tour view on the left is a dynamic and interactive scatterplot. Brushing on the tour view is activated with the shift key plus a mouse drag. By default it will highlight corresponding points in the xy view and pause the animation.
• The xy view on the right is an interactive scatterplot. Brushing on the xy view will highlight points in the tour view and is activated via a mouse drag, the type of highlighting depends on the brush mode selected.
• There is a play button, that when pressed will start the tour.
• The half range which is the maximum squared Euclidean distance between points in the tour view. The half range is a scale factor for projections and can be thought of as a way of zooming in and out on points. It can be dynamically modified by scrolling (via a mouse-wheel). To reset double click the tour view.
• The legend can be toggled to highlight groups of points with shift+mouse-click. Multiple groups can be selected in this way. To reset double click the legend title.

Value

After pressing the Done button on the interface, a list of artefacts is returned to the R session.

• selected_basis: A matrix of the current projection
If (interactive()) {
    # tour the first ten columns of the fake tree data and link to the
    # another layout based on t-SNE
    # loads the default interface
    if (requireNamespace("Rtsne", quietly = TRUE)) {
        set.seed(2020)
        tsne <- Rtsne::Rtsne(dplyr::select(fake_trees, dplyr::starts_with("dim")))
        tsne_df <- data.frame(tsneX = tsne$Y[, 1], tsneY = tsne$Y[, 2])
        limn_tour_link(
            tsne_df,
            fake_trees,
            cols = dim1:dim10,
            color = branches
        )
        # assigning to an object will return a list of artefacts after clicking
        # done in the upper right hand corner
        res <- limn_tour_link(tsne_df, fake_trees, cols = dim1:dim10, color = branches)
    }
}
Details

These functions are designed to alter the resulting projection after basis generation with the `tourr` and will change how the projections are animated with `limn_tour()` and `limn_tour_link()`. For `morph_center()` the projection is centered and then scaled by the half range, while `morph_identity()` only scales by half range. `morph_radial()` is an implementation of the burning sage algorithm available in `tourr::display_sage()`.

Value

A matrix with dimensions the same as `proj`.

Examples

```r
proj <- matrix(rnorm(20), ncol = 2)
half_range <- compute_half_range(proj)
morph_center(proj, half_range)
morph_identity(proj, half_range)
morph_radial(proj, half_range, p_eff = 2)
```

---

pdfsense  
Parton distribution function sensitivity experiments

Description

Data from Wang et al., 2018 to compare embedding approaches to a tour path.

Usage

```r
pdfsense
```

Format

An object of class `data.frame` with 2808 rows and 62 columns.

Details

Data were obtained from CT14HERA2 parton distribution function fits as used in Laa et al., 2018. There are 28 directions in the parameter space of parton distribution function fit, each point in the variables labelled X1-X56 indicate moving +- 1 standard deviation from the 'best' (maximum likelihood estimate) fit of the function. Each observation has all predictions of the corresponding measurement from an experiment.

(see table 3 in that paper for more explicit details).

The remaining columns are:

- InFit: A flag indicating whether an observation entered the fit of CT14HERA2 parton distribution function
- Type: First number of ID
• ID: contains the identifier of experiment, 1XX/2XX/5XX corresponts to Deep Inelastic Scattering (DIS) / Vector Boson Production (VBP) / Strong Interaction (JET). Every ID points to an experimental paper.
• pt: the per experiment observational id
• x,mu: the kinematics of a parton. x is the parton momentum fraction, and mu is the factorisation scale.

Source
http://www.physics.smu.edu/botingw/PDFsense_web_histlogy/

References
Index

* datasets
  fake_trees, 4
  pdfsense, 10

clamp, 2
clamp(), 6, 8
clamp_robust(clamp), 2
clamp_sd(clamp), 2
clamp_standardize(clamp), 2
compute_half_range, 3
compute_half_range(), 7
compute_proj_dist, 4
ake_trees, 4

datasets

identity(), 6, 8
limn_pal_tableau10, 5
limn_pal_tableau20
  (limn_pal_tableau10), 5
limn_tour, 6
limn_tour(), 5, 10
limn_tour_link, 7
limn_tour_link(), 7, 10

limn_pal_tableau20

morph_center, 9
morph_center(), 6–8, 10
morph_identity(morph_center), 9
morph_identity(), 10
morph_radial(morph_center), 9
morph_radial(), 10

pdfsense, 10

shiny::runGadget(), 6, 8
tidyselect::starts_with(), 6, 8
tourr, 10
tourr::display_sage(), 9, 10
tourr::grand_tour(), 6, 8
tourr::guided_tour(), 6, 8