Package ‘detourr’

October 13, 2022

Type Package
Title Portable and Performant Tour Animations
Version 0.1.0
Description Provides 2D and 3D tour animations as HTML widgets. The user can interact with the widgets using orbit controls, tooltips, brushing, and timeline controls. Linked brushing is supported using ‘crosstalk’, and widgets can be embedded in Shiny apps or HTML documents.
License MIT + file LICENSE
Encoding UTF-8
LazyData true
Imports tourr, tibble, htmlwidgets, tidyselect, rlang, purrr, viridisLite, grDevices
RoxygenNote 7.2.0
Suggests testthat (>= 3.0.0), crosstalk, shiny, htmltools, knitr, dplyr, liminal, rmarkdown, geozoo
Config/testthat/edition 3
URL https://casperhart.github.io/detourr/
BugReports https://github.com/casperhart/detourr/issues
Depends R (>= 4.1)
NeedsCompilation no
Author Casper Hart [aut, cre], Earo Wang [aut, ths] (<https://orcid.org/0000-0001-6448-5260>)
Maintainer Casper Hart <casperhart93@gmail.com>
Repository CRAN
Date/Publication 2022-06-20 08:00:02 UTC


R topics documented:

detour ................................................................. 2
detour-shiny ......................................................... 3
is_detour ............................................................ 3
mnist_embeddings ................................................ 4
show_sage ............................................................ 5
show_scatter ........................................................ 7
show_scatter_internal ........................................... 8
show_slice .......................................................... 10
tour_aes ............................................................ 12
tour_path ............................................................ 12

Index 14

detour Initiate a detour visualisation

Description

This function initialises a detour object which, along with the tour_path and show_functions is used to build a detour visualisation.

Usage

detour(.data, mapping)

Arguments

.data a data frame, tibble, or crosstalk::SharedData object
mapping a mapping of data columns to aesthetic values using the tour_aes function.
The only required aesthetic is projection, which determines which columns are used to generate the tour path and supports tidy selection.

Value

A detour object containing information about the tour path and its parameters

Examples

detour(tourr::flea, tour_aes(projection = -species, colour = species)) |> 
tour_path(grand_tour(3), fps = 60) |> 
show_scatter(alpha = 0.7, axes = FALSE)
Shiny bindings for detourr

Description
Output and render functions for using detourr with shiny. The output function used must match both the display method and tour dim used, or it will lead to strange behaviour.

Usage

\begin{verbatim}
displayScatter3dOutput(output_id, width = "100\%", height = "400px")
displayScatter2dOutput(output_id, width = "100\%", height = "400px")
shinyRenderDisplayScatter2d(expr, env = parent.frame(), quoted = FALSE)
shinyRenderDisplayScatter3d(expr, env = parent.frame(), quoted = FALSE)
\end{verbatim}

Arguments

output_id output variable to read from
width, height Must be a valid CSS unit (like "100\%", "400px", "auto") or a number, which will be coerced to a string and have "px" appended.
expr an expression that generates a detourr widget
env The environment in which to evaluate expr.
quoted Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

Value
An output or render function that enables the use of the widget within shiny applications

Test for detour-ness

Description
Test for detour-ness

Usage
\begin{verbatim}
is_detour(x)
\end{verbatim}
Arguments

x an object

Value

TRUE or FALSE

Description

Two datasets are available: mnist_embeddings_8d contains 8-dimensional embedding vectors and mnist_embeddings_32d contains 32-dimensional embedding vectors.

The neural network that produced these embeddings was created using TensorFlow (Abadi et al. (2016)) with a variation of the code found in this example: https://www.tensorflow.org/addons/tutorials/losses_triplet

Usage

mnist_embeddings_32d

mnist_embeddings_8d

Format

An object of class tbl_df (inherits from tbl.data.frame) with 10000 rows and 34 columns.
An object of class tbl_df (inherits from tbl.data.frame) with 10000 rows and 10 columns.

Details

A data frame with 10,000 rows and p variables:

- id: sequential ID or row number of the image
- label: the digit 0, 1, ..., 9
- X1–Xp: elements 1–p of the embedding vector

References

show_sage

2D and 3D Sage Tour Display

Description

An implementation of the Sage Tour described in Laa et al., 2021. It uses a radial transformation on the projected data so that the relative volume is preserved when the data is projected. I.e. a uniform distribution in the original space will remain uniformly distributed in the projected space. Includes both 2D and 3D variations.

Usage

```
show_sage(
  x,
  ..., 
  palette = viridis,
  center = TRUE,
  axes = TRUE,
  edges = NULL,
  paused = TRUE,
  scale_factor = NULL,
  gamma = 1,
  R = NULL
)
```

Arguments

- **x**
  - a detour object
- **...**
  - used to support aesthetic parameters for the plot, including
    - size: point size, defaults to 1
    - alpha: point opacity, defaults to 1
    - background_colour: defaults to "white"
- **palette**
  - Colour palette to use with the colour aesthetic. Can be:
    - A character vector of R colours. This should match the number of levels of the colour aesthetic, or the number of bins to use for continuous colours.
    - A function which takes the number of colours to use as input and returns a character vector of colour names and / or hex values as output.
- **center**
  - If TRUE, center the projected data to (0, 0, 0).
- **axes**
  - Can be one of:
    - TRUE draw axes and use column names for axis labels
    - FALSE do not draw axes or labels
    - NULL draw axes with no labels
    - An unnamed vector of labels with the same length as cols
    - A named vector in the form c("h" = "head"), where head is renamed to h
show_sage

edges  A two column numeric matrix giving indices of ends of lines.
paused whether the widget should be initialised in the ‘paused’ state
scale_factor used as a multiplier for the point coordinates so they are displayed on a sensible
        range. Defaults to the reciprocal of maximum distance from a point to the origin.
gamma  the gamma parameter for scaling the effective dimensionality for the sage tour
        radial transformation. defaults to 1
R      scale for the radial transformation. Defaults to scale_factor times the maximum
distance from the origin to each row of data. If the default scale_factor is used this will result in R=1. Because the
        R and scale_factor parameters interact with one another, it is recommended to leave scale_factor at its default
        value, and modify R if needed.

Details

This display method produces an interactive scatterplot animation which supports both 2D and 3D tours. Linked selection and filtering is also supported using crosstalk. The set of interactive controls available are:

- A timeline with a play / pause button and indicators at the position of each basis used. The basis indicators can be hovered with the mouse to show the index of the basis, or clicked to jump to that basis. The timeline also allows for clicking and dragging of the scrubber to move to any individual frame of the animation.
- Orbit controls. For the 2D variant, this allows the projection to be rotated by clicking and dragging from left to right. For the 3D variant, full orbit controls are available by clicking and dragging. For both orbit and pan controls, the scroll wheel can be used to zoom.
- Pan controls, which work similarly to orbit controls but move the camera laterally / vertically rather than rotating
- Resetting of the orbit and pan controls
- Selection and highlighting. Multiple selection is possible by using the shift key
- Colouring / brushing of highlighted points

Value

An object of class htmlwidget

References


See Also

show_scatter

Examples

detour(tourr::flea, tour_aes(projection = -species, colour = species)) |> 
tour_path(grand_tour(3), fps = 60) |> 
show_sage(gamma = 2)
**show_scatter**

**2D and 3D Scatter Plot Display for Tours**

**Description**

Display method for a high performance 2D or 3D scatter plot. Performance is achieved through the use of Three.js / WebGL, and the 2D or 3D variant is selected automatically based on the tour generator provided.

**Usage**

```r
show_scatter(
  x, 
  ..., 
  palette = viridis, 
  center = TRUE, 
  axes = TRUE, 
  edges = NULL, 
  paused = TRUE, 
  scale_factor = NULL
)
```

**Arguments**

- `x` a `detour` object
- `...` used to support aesthetic parameters for the plot, including
  - size: point size, defaults to 1
  - alpha: point opacity, defaults to 1
  - background_colour: defaults to "white"
- `palette` Colour palette to use with the colour aesthetic. Can be:
  - A character vector of R colours. This should match the number of levels of the colour aesthetic, or the number of bins to use for continuous colours.
  - A function which takes the number of colours to use as input and returns a character vector of colour names and/or hex values as output.
- `center` If TRUE, center the projected data to (0, 0, 0).
- `axes` Can be one of:
  - TRUE draw axes and use column names for axis labels
  - FALSE do not draw axes or labels
  - NULL draw axes with no labels
  - An unnamed vector of labels with the same length as cols
  - A named vector in the form `c("h" = "head")`, where head is renamed to h
- `edges` A two column numeric matrix giving indices of ends of lines.
- `paused` whether the widget should be initialised in the 'paused' state
- `scale_factor` used as a multiplier for the point coordinates so they are displayed on a sensible range. Defaults to the reciprocal of maximum distance from a point to the origin.
Details

This display method produces an interactive scatterplot animation which supports both 2D and 3D tours. Linked selection and filtering is also supported using crosstalk. The set of interactive controls available are:

- A timeline with a play / pause button and indicators at the position of each basis used. The basis indicators can be hovered with the mouse to show the index of the basis, or clicked to jump to that basis. The timeline also allows for clicking and dragging of the scrubber to move to any individual frame of the animation.
- Orbit controls. For the 2D variant, this allows the projection to be rotated by clicking and dragging from left to right. For the 3D variant, full orbit controls are available by clicking and dragging. For both orbit and pan controls, the scroll wheel can be used to zoom.
- Pan controls, which work similarly to orbit controls but move the camera laterally / vertically rather than rotating
- Resetting of the orbit and pan controls
- Selection and highlighting. Multiple selection is possible by using the shift key
- Colouring / brushing of highlighted points

Value

An object of class htmlwidget

Examples

```r
detour(tourr::flea, tour_aes(projection = -species, colour = species)) |> tour_path(grand_tour(3), fps = 60) |> show_scatter(alpha = 0.7, axes = FALSE)
```

show_scatter_internal

Internal method for 2D and 3D Scatter Plot Display

Description

Internal method for 2D and 3D Scatter Plot Display

Usage

```r
show_scatter_internal(
  x,
  ..., 
  palette = viridisLite::viridis,
  center = TRUE,
  axes = TRUE,
  edges = NULL,
  paused = TRUE,
  scale_factor = NULL
)
```
show_scatter_internal

Arguments

- **x**
  a detour object

- **...**
  used to support aesthetic parameters for the plot, including
    - **size**: point size, defaults to 1
    - **alpha**: point opacity, defaults to 1
    - **background_colour**: defaults to "white"

- **palette**
  Colour palette to use with the colour aesthetic. Can be:
    - A character vector of R colours. This should match the number of levels of the colour aesthetic, or the number of bins to use for continuous colours.
    - A function which takes the number of colours to use as input and returns a character vector of colour names and/or hex values as output.

- **center**
  If TRUE, center the projected data to (0, 0, 0).

- **axes**
  Can be one of:
    - TRUE draw axes and use column names for axis labels
    - FALSE do not draw axes or labels
    - NULL draw axes with no labels
    - An unnamed vector of labels with the same length as cols
    - A named vector in the form c("h" = "head"), where head is renamed to h

- **edges**
  A two column numeric matrix giving indices of ends of lines.

- **paused**
  Whether the widget should be initialised in the 'paused' state

- **scale_factor**
  Used as a multiplier for the point coordinates so they are displayed on a sensible range. Defaults to the reciprocal of maximum distance from a point to the origin.

Details

This display method produces an interactive scatterplot animation which supports both 2D and 3D tours. Linked selection and filtering is also supported using crosstalk. The set of interactive controls available are:

- A timeline with a play / pause button and indicators at the position of each basis used. The basis indicators can be hovered with the mouse to show the index of the basis, or clicked to jump to that basis. The timeline also allows for clicking and dragging of the scrubber to move to any individual frame of the animation.

- Orbit controls. For the 2D variant, this allows the projection to be rotated by clicking and dragging from left to right. For the 3D variant, full orbit controls are available by clicking and dragging. For both orbit and pan controls, the scroll wheel can be used to zoom.

- Pan controls, which work similarly to orbit controls but move the camera laterally / vertically rather than rotating

- Resetting of the orbit and pan controls

- Selection and highlighting. Multiple selection is possible by using the shift key

- Colouring / brushing of highlighted points

Value

An object of class htmlwidget
show_slice  

2D and 3D Slice Display for Tours

Description

An implementation of the Slice Tour described in Laa et al., 2020. Points close to the projection plane are highlighted, and those far away are faded out.

Usage

```r
show_slice(
  x,
  ..., 
  palette = viridis, 
  center = TRUE,
  axes = TRUE,
  edges = NULL, 
  paused = TRUE, 
  scale_factor = NULL, 
  slice_relative_volume = 0.1, 
  anchor = NULL
)
```

Arguments

- **x**: a detour object
- **...**: used to support aesthetic parameters for the plot, including
  - `size`: point size, defaults to 1
  - `alpha`: point opacity, defaults to 1
  - `background_colour`: defaults to "white"
- **palette**: Colour palette to use with the colour aesthetic. Can be:
  - A character vector of R colours. This should match the number of levels of the colour aesthetic, or the number of bins to use for continuous colours.
  - A function which takes the number of colours to use as input and returns a character vector of colour names and / or hex values as output.
- **center**: If TRUE, center the projected data to (0, 0, 0).
- **axes**: Can be one of:
  - `TRUE` draw axes and use column names for axis labels
  - `FALSE` do not draw axes or labels
  - `NULL` draw axes with no labels
  - An unnamed vector of labels with the same length as cols
  - A named vector in the form c("h" = "head"), where head is renamed to h
- **edges**: A two column numeric matrix giving indices of ends of lines.
show_slice

paused whether the widget should be initialised in the 'paused' state

scale_factor used as a multiplier for the point coordinates so they are displayed on a sensible range. Defaults to the reciprocal of maximum distance from a point to the origin.

slice_relative_volume number default 0.1. Controls the relative volume of the slice and thus the number of points which are highlighted. This is an approximate value and is only accurate for values ≤ 1

anchor anchor vector of length p to offset the projection plane when calculating the distance from each point to the projection plane.

Details

This display method produces an interactive scatterplot animation which supports both 2D and 3D tours. Linked selection and filtering is also supported using crosstalk. The set of interactive controls available are:

- A timeline with a play / pause button and indicators at the position of each basis used. The basis indicators can be hovered with the mouse to show the index of the basis, or clicked to jump to that basis. The timeline also allows for clicking and dragging of the scrubber to move to any individual frame of the animation.
- Orbit controls. For the 2D variant, this allows the projection to be rotated by clicking and dragging from left to right. For the 3D variant, full orbit controls are available by clicking and dragging. For both orbit and pan controls, the scroll wheel can be used to zoom.
- Pan controls, which work similarly to orbit controls but move the camera laterally / vertically rather than rotating
- Resetting of the orbit and pan controls
- Selection and highlighting. Multiple selection is possible by using the shift key
- Colouring / brushing of highlighted points

Value

An object of class htmlwidget

References


See Also

show_scatter

Examples

```r
x <- geozoo::torus(p = 4, n = 10000)$points |
  tibble::as_tibble(.name_repair = "unique")

detour(x, tour_aes(projection = everything())) |
```
tour_path(grand_tour(2)) |> show_slice(slice_relative_volume = 0.1)

tour_aes

Aesthetic mapping for tours

Description
Aesthetic mapping for tours describing how variables in the data are mapped to visual properties of the tour animation.

Usage
tour_aes(...)

Arguments
... list of name-value pairs in the form 'aesthetic = variable'. Variables are evaluated using tidyselect syntax.

Value
a list of quosures

Examples
detour(tourr::flea, tour_aes(projection = -species, colour = species)) |>
tour_path(grand_tour(3), fps = 60) |>
show_scatter(alpha = 0.7, axes = FALSE)

tour_path Generate a tour path for a detour object

Description
This function takes a detour object as an input, and generates a sequence of projection matrices for the tour. The return value is another detour object with the tour path and other metadata attached. This can then be passed to a show_* function to generate the detour visualisation.

Usage
tour_path(
  x,
  tour_path = grand_tour(2),
  start = NULL,
  aps = 1,
  fps = 30,
  max_bases = 10
)

tour_path

Arguments

- \(x\) a detour object
- \(\text{tour\_path}\) tour path generator, defaults to 2d grand tour
- \(\text{start}\) projection to start at, if not specified, uses default associated with tour path
- \(\text{aps}\) target angular velocity (in radians per second)
- \(\text{fps}\) target frames per second
- \(\text{max\_bases}\) the maximum number of bases to generate

Value

A detour object containing information about the tour path and its parameters
Index

* datasets
  - mnist_embeddings, 4

  detour, 2
  detour-shiny, 3
  displayScatter2dOutput (detour-shiny), 3
  displayScatter3dOutput (detour-shiny), 3

  is_detour, 3

  mnist_embeddings, 4
  mnist_embeddings_32d (mnist_embeddings), 4
  mnist_embeddings_8d (mnist_embeddings), 4

  shinyRenderDisplayScatter2d (detour-shiny), 3
  shinyRenderDisplayScatter3d (detour-shiny), 3

  show_sage, 5
  show_scatter, 7
  show_scatter_internal, 8
  show_slice, 10

  tour_aes, 12
  tour_path, 12