

Package ‘causalimages’

February 22, 2023

Title causalimages: R Package for Causal Inference with Earth Observation, Bio-medical, and Social Science Images

Version 2.0

Authors 'Connor Jerzak <connor.jerzak@gmail.com> [aut, cre], Fredrik Johansson <fredrik.johansson@chalmers.se> [aut], Adel Daoud <adel.daoud@liu.se> [aut]'

Description R Package for causal inference with earth observation, bio-medical, and social science images and image sequences (i.e., videos)

Depends R (>= 3.3.3)

License

Creative Commons Attribution-Noncommercial-No Derivative Works 4.0, for academic use only.

Encoding UTF-8

LazyData true

Maintainer 'Connor Jerzak' <connor.jerzak@gmail.com>

Imports tensorflow

RoxygenNote 7.2.1

R topics documented:

AnalyzeImageHeterogeneity	1
SimulateImageSystem	2
Index	4

AnalyzeImageHeterogeneity

Decompose treatment effect heterogeneity by image

Description

Implements the image heterogeneity decomposition analysis of Jerzak, Johansson, and Daoud (2023).

Usage

AnalyzeImageHeterogeneity(obsW, obsY, acquireImageFxn, ...)

Arguments

obsW 'DAG'.
 nMonte_predictive
 (default = '10L') An integer specifying how many Monte Carlo iterations to use in the calculation approximate posterior means (e.g., mean cluster probabilities).

Value

A list consiting of

- Items.

References

- Connor T. Jerzak, Fredrik Johansson, Adel Daoud. Image-based Treatment Effect Heterogeneity. Forthcoming in *Proceedings of the Second Conference on Causal Learning and Reasoning (CLearR), Proceedings of Machine Learning Research (PMLR)*, 2023.

Examples

```
#set seed
set.seed(1)

#Geneate data
x <- rnorm(100)
```

SimulateImageSystem	<i>Simulate causal systems involving images</i>
---------------------	---

Description

This function generates simulated causal structures using images. It is currently under construction.

Usage

```
SimulateImageSystem(...)
```

Arguments

dag	(<i>character string</i>) An input DAG specifying causal structure. This input should be of the form ' <i>i</i> -> <i>t</i> , <i>i</i> -> <i>y</i> , <i>t</i> -> <i>y</i> , ...'. Currently, only one node in a DAG can be an image (this should be labeled " <i>i</i> "). The non-image nodes can have arbitrary string labels. The image can be a confounder, effect moderator, effect mediator. If the image is to be used as a moderator, use the notation, <i>t</i> - <i>i</i> > <i>y</i> .
...	(<i>optional</i>) In estimation mode, users input the data matrices associated with the non-image nodes of <i>DAG</i> and image node <i>i</i> . For example, if <i>x</i> is a DAG node, users must, in estimation mode, supply data to <i>x</i> in a form that can be coerced to a tensor.
treatment	(<i>character string, optional</i>) In estimation mode, users specify the treatment variable here. If <i>treatment</i> is specified, users must provide other data inputs to the DAG (see ...).

<code>image_pool</code>	(<i>character string, optional</i>) The path to where analysis specific images are located. This can be specified both in simulation and estimation mode. If not specified, the simulation uses a pool of Landsat images from Nigeria.
<code>analysis_level</code>	(<i>character string, default is 'scene'</i>) Defines the unit of analysis used in the simulation framework. This is ignored in estimation mode, where the unit of analysis is inferred from the data dimensions.
<code>control</code>	(<i>list</i>) A list containing control parameters in the data generating process.

Value

A list:

- In *simulation mode*, the function returns a list with as many elements as unique nodes in DAG. Each element represents the simulated data.
- In *estimation mode*, the function returns an estimated treatment effect with 95% confidence intervals.

References

- Connor T. Jerzak, Fredrik Johansson, Adel Daoud. Image-based Treatment Effect Heterogeneity. Forthcoming in *Proceedings of the Second Conference on Causal Learning and Reasoning (CLEaR), Proceedings of Machine Learning Research (PMLR)*, 2023.

Examples

```
#set seed
set.seed(1)

# Simulation mode
#simulatedData <- causalimage('r->i, i->t, t->y, r->y')
#print(names(simulatedData))

# Estimation mode
#estimatedResults <- causalimage('r->i, i->t, t->y, r->y', y=y, r=r, y=y', treatment='t')
#print( estimatedResults )
```

Index

AnalyzeImageHeterogeneity, [1](#)

SimulateImageSystem, [2](#)