

Package ‘einet’

July 22, 2025

Type Package

Title Effective Information and Causal Emergence

Version 0.1.0

Description Methods and utilities for causal emergence.

Used to explore and compute various information theory metrics for networks, such as effective information, effectiveness and causal emergence.

License MIT + file LICENSE

URL <https://github.com/travisbyrum/einet>

BugReports <https://github.com/travisbyrum/einet/issues>

Depends R (>= 3.2.0)

Encoding UTF-8

LazyData true

Imports assertthat, igraph, magrittr, shiny, entropy

Suggests testthat, RColorBrewer, knitr, rmarkdown, bench

VignetteBuilder knitr

RoxygenNote 7.0.2

NeedsCompilation no

Author Travis Byrum [aut, cre],

Anshuman Swain [aut],

Brennan Klein [aut],

William Fagan [aut]

Maintainer Travis Byrum <tbyrum@terpmail.umd.edu>

Repository CRAN

Date/Publication 2020-04-23 17:20:03 UTC

Contents

causal_emergence	2
check_network	3

create_macro	3
effective_information	4
einet	5
karate	5
mb	6
run_example	6
stationary	7
update_blanket	7
Index	8

causal_emergence	<i>Causal Emergence</i>
------------------	-------------------------

Description

Given a microscale network, G, this function iteratively checks different coarse-grainings to see if it finds one with higher effective information.

Usage

causal_emergence(x, ...)

Arguments

- x igraph or matrix object.
- ... Span, and threshold parameters

Value

- A list with letters and numbers.
- g_micro - Graph of original micro-scale network.
 - g_macro - Graph of macro-scale network.
 - mapping - list mapping from micro to macro scales giving the largest increase in effective information.
 - ei_macro - Effective information of macro scale network.
 - ei_micro - Effective information of micro scale network.
 - ce - Numerical value for causal emergence.

Examples

```
graph <- matrix(
  cbind(
    c(0.0, 1.0, 0.0, 0.0),
    c(0.0, 0.0, 1.0, 0.0),
    c(0.0, 0.0, 0.0, 1.0),
    c(0.0, 0.0, 0.0, 0.0)
  ),
  nrow = 4
) %>%
  igraph::graph.adjacency(mode = "directed")

causal_emergence(graph)
```

check_network	<i>Check Graph Network</i>
---------------	----------------------------

Description

check_network returns processed graph.

Usage

```
check_network(graph)
```

Arguments

graph igraph

Details

This is a pre-processing function that turns raw input into directed networks with edge weights.

create_macro	<i>create_macro</i>
--------------	---------------------

Description

Coarse-grains a network according to the specified macro_mapping and the types of macros that each macro is associated with.

Usage

```
create_macro(graph, mapping, macro_types, ...)
```

Arguments

graph	igraph
mapping	List mapping from micro to macro nodes.
macro_types	List of node distribution types.
...	Passed arguments.

Value

Directed igraph graph object corresponding to a coarse-grained network according to the mapping of micro nodes onto macro nodes, given by mapping.

effective_information *Effective Information*

Description

Calculates the effective information (EI) of a network, G , according to the definition provided in Klein & Hoel, 2019. Here, we subtract the average entropies of the out-weights of nodes in a network, WOUT_average from the entropy of the average out-weights in the network, WIN_entropy.

Usage

```
effective_information(graph, effectiveness = FALSE)
```

Arguments

graph	igraph or matrix object.
effectiveness	Logical indicating whether or not to return network effectiveness.

Value

Numeric value indicating the effective information of the network.

Examples

```
graph <- matrix(
  cbind(
    c(0.0, 1.0, 0.0, 0.0),
    c(0.0, 0.0, 1.0, 0.0),
    c(0.0, 0.0, 0.0, 1.0),
    c(0.0, 0.0, 0.0, 0.0)
  ),
  nrow = 4
) %>%
  igraph::graph.adjacency(mode = "directed")

effective_information(graph)
```

einet*einet: Uncertainty and causal emergence in complex networks.*

Description

for calculating effective information in networks. This can then be used to search for macroscale representations of a network such that the coarse grained representation has more effective information than the microscale, a phenomenon known as causal emergence.

Author(s)

Maintainer: Travis Byrum <tbyrum@terpmail.umd.edu>

Authors:

- Anshuman Swain <answain@terpmail.umd.edu>
- Brennan Klein <klein.br@northeastern.edu>
- William Fagan <bfagan@umd.edu>

See Also

Useful links:

- <https://github.com/travisbyrum/einet>
- Report bugs at <https://github.com/travisbyrum/einet/issues>

karate*Zachary's karate club*

Description

Social network data of university karate club. Used for causal emergence benchmarking and testing.

Usage

karate

Format

Igraph object with 78 edges.

Source

<http://www-personal.umich.edu/~mejn/netdata/>

mb	Create Markov Blanket
----	-----------------------

Description

Given a graph and a specified vector of internal node(s), returns the parents, the children, and the parents of the children of the internal node(s).

Usage

```
mb(graph, nodes = igraph::V(graph))
```

Arguments

graph	igraph or matrix object.
nodes	Numeric vector of vertices.

Value

A list of node descendants, parents, and neighbors.

run_example	Start shiny app
-------------	-----------------

Description

This starts an example shiny app that allows for user inputted graph objects.

Usage

```
run_example()
```

stationary	<i>Stationary Distribution</i>
------------	--------------------------------

Description

Gives a stationary probability vector of a given network.

Usage

```
stationary(graph, zero_cutoff = 1e-10)
```

Arguments

graph	igraph or matrix object.
zero_cutoff	Numeric threshold for zero value.

Value

A numeric vector corresponding to stationary distribution.

update_blanket	<i>Update Markov Blanket</i>
----------------	------------------------------

Description

Update Markov Blanket

Usage

```
update_blanket(blanket, removal = NULL)
```

Arguments

blanket	List of previous markov blanket.
removal	Numeric vector for node removal.

Index

* **datasets**

karate, [5](#)

causal_emergence, [2](#)

check_network, [3](#)

create_macro, [3](#)

effective_information, [4](#)

einet, [5](#)

einet-package (einet), [5](#)

karate, [5](#)

mb, [6](#)

run_example, [6](#)

stationary, [7](#)

update_blanket, [7](#)