

Package ‘NetworkInference’

December 1, 2017

Type Package

Title Inferring Latent Diffusion Networks

Version 1.1.2

Date 2017-12-01

Description This is an R implementation of the netinf algorithm (Gomez Rodriguez, Leskovec, and Krause, 2010)<doi:10.1145/1835804.1835933>. Given a set of events that spread between a set of nodes the algorithm infers the most likely stable diffusion network that is underlying the diffusion process.

License MIT + file LICENSE

Imports Rcpp (>= 0.12.5), assertthat, checkmate, ggplot2, ggrepel, stats

LinkingTo Rcpp, RcppProgress

BugReports <https://github.com/desmarais-lab/NetworkInference/issues>

Suggests testthat, knitr, rmarkdown, pander, igraph, utils

RoxygenNote 6.0.1

SystemRequirements C++11

LazyData true

VignetteBuilder knitr

NeedsCompilation yes

Author Fridolin Linder [aut, cre],
Bruce Desmarais [ctb]

Maintainer Fridolin Linder <fridolin.linder@gmail.com>

Repository CRAN

Date/Publication 2017-12-01 21:18:48 UTC

R topics documented:

as.data.frame.cascade	2
as.matrix.cascade	3

as_cascade_long	4
as_cascade_wide	5
cascades	6
count_possible_edges	6
is.cascade	7
is.diffnet	8
netinf	8
NetworkInference	9
plot.cascade	10
plot.diffnet	11
policies	12
simulate_cascades	13
simulate_rnd_cascades	14
subset_cascade	15
subset_cascade_time	15
summary.cascade	16
validation	17
Index	18

as.data.frame.cascade *Convert a cascade object to a data frame*

Description

Generates a data frame containing the cascade information in the cascade object.

Usage

```
## S3 method for class 'cascade'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	Cascade object to convert.
row.names	NULL or a character vector giving the row names for the data frame. Missing values are not allowed.
optional	logical. If TRUE, setting row names and converting column names (to syntactic names: see <code>make.names</code>) is optional. (Not supported)
...	Additional arguments passed to <code>data.frame</code> .

Value

A data frame with three columns. Containing 1) The names of the nodes ("node_name") that experience an event in each cascade, 2) the event time ("event_time") of the corresponding node, 3) the cascade identifier "cascade_id".

Examples

```
data(cascades)
as.data.frame(cascades)
```

as.matrix.cascade	<i>Convert a cascade object to a matrix</i>
-------------------	---

Description

Generates a [matrix](#) containing the cascade information in the cascade object in wide format. Missing values are used for nodes that do not experience an event in a cascade.

Usage

```
## S3 method for class 'cascade'
as.matrix(x, ...)
```

Arguments

x	cascade object to convert.
...	additional arguments to be passed to or from methods. (Currently not supported.)

Value

A matrix containing all cascade information in wide format. That is, each row of the matrix corresponds to a node and each column to a cascade. Cell entries are event times. Censored nodes have NA for their entry.

Examples

```
data(cascades)
as.matrix(cascades)
```

as_cascade_long	<i>Transform long data to cascade</i>
-----------------	---------------------------------------

Description

Create a cascade object from data in long format.

Usage

```
as_cascade_long(data, cascade_node_name = "node_name",
  event_time = "event_time", cascade_id = "cascade_id", node_names = NULL)
```

Arguments

data	data.frame , containing the cascade data with column names corresponding to the arguments provided to cascade_node_names, event_time and cascade_id.
cascade_node_name	character, column name of data that specifies the node names in the cascade.
event_time	character, column name of data that specifies the event times for each node involved in a cascade.
cascade_id	character, column name of the cascade identifier.
node_names	character, factor or numeric vector containing the names for each node. Optional. If not provided, node names are inferred from the cascade data.

Details

Each row of the data describes one event in the cascade. The data must contain at least three columns:

1. Cascade node name: The identifier of the node that experiences the event.
2. Event time: The time when the node experiences the event. Note that if the time column is of class date or any other special time class, it will be converted to an integer with 'as.numeric()'.
3. Cascade id: The identifier of the cascade that the event pertains to.

The default names for these columns are node_name, event_time and cascade_id. If other names are used in the data object the names have to be specified in the corresponding arguments (see argument documentation)

Value

An object of class cascade. This is a list containing three (named) elements:

1. "node_names" A character vector of node names.
2. "cascade_nodes" A list with one character vector per cascade containing the node names in order of the events.
3. "cascade_times" A list with one element per cascade containing the event times for the nodes in "cascade_names".

Examples

```
df <- simulate_rnd_cascades(10, n_nodes = 20)
cascades <- as_cascade_long(df)
is.cascade(cascades)
```

as_cascade_wide	<i>Transform wide data to cascade</i>
-----------------	---------------------------------------

Description

Create a cascade object from data in wide format.

Usage

```
as_cascade_wide(data, node_names = NULL)
```

Arguments

data	data.frame or matrix , rows corresponding to nodes, columns to cascades. Matrix entries are the event times for each node, cascade pair. Missing values indicate censored observations, that is, nodes that did not have an event). Specify column and row names if cascade and node ids other than integer sequences are desired. Note that, if the time column is of class date or any other special time class, it will be converted to an integer with 'as.numeric()'.
node_names	character, factor or numeric vector, containing names for each node. Optional. If not provided, node names are inferred from the provided data.

Details

If data is in wide format, each row corresponds to a node and each column to a cascade. Each cell indicates the event time for a node - cascade combination. If a node did not experience an event for a cascade (the node is censored) the cell entry must be NA.

Value

An object of class cascade. This is a list containing three (named) elements:

1. "node_names" A character vector of node names.
2. "cascade_nodes" A list with one character vector per cascade containing the node names in order of the events.
3. "cascade_times" A list with one element per cascade containing the event times for the nodes in "cascade_names".

Examples

```
data("policies")
cascades <- as_cascade_wide(policies)
is.cascade(cascades)
```

cascades	<i>Example cascades</i>
----------	-------------------------

Description

An example dataset of 31 nodes and 54 cascades. From the original netinf implementation in SNAP.

Usage

```
data(cascades)
```

Format

An object of class cascade containing 4 objects

node_names Character node names

cascade_nodes A list of integer vectors. Each containing the names of the nodes infected in this cascades in the order of infection

cascade_times A list of numeric vectors. Each containing the infection times for the corresponding nodes in cascade_nodes

Source

<https://github.com/snap-stanford/snap/blob/master/examples/netinf/example-cascades.txt>

count_possible_edges	<i>Count the number of possible edges in the dataset</i>
----------------------	--

Description

Across all cascades, count the edges that are possible. An edge from node u to node v is only possible if in at least one cascade u experienced an event before v .

Usage

```
count_possible_edges(cascades)
```

Arguments

cascades Object of class cascade containing the data.

Value

An integer count.

Examples

```
data(cascades)
count_possible_edges(cascades)
```

is.cascade *Is the object of class cascade?*

Description

Is the object of class cascade?

Usage

```
is.cascade(object)
```

Arguments

object the object to be tested.

Value

TRUE if object is a cascade, FALSE otherwise.

Examples

```
data(cascades)
is.cascade(cascades)
# > TRUE
is.cascade(1)
# > FALSE
```

is.diffnet	<i>Is the object of class diffnet?</i>
------------	--

Description

Tests if an object is of class diffnet. The class diffnet is appended to the object returned by [netinf](#) for dispatch of appropriate plotting methods.

Usage

```
is.diffnet(object)
```

Arguments

object the object to be tested.

Value

TRUE if object is a diffnet, FALSE otherwise.

Examples

```
data(cascades)
result <- netinf(cascades, n_edges = 6, lambda = 1)
is.diffnet(result)
```

netinf	<i>Infer latent diffusion network</i>
--------	---------------------------------------

Description

Infer a network of diffusion ties from a set of cascades. Each cascade is defined by pairs of node ids and infection times.

Usage

```
netinf(cascades, trans_mod = "exponential", n_edges, lambda, quiet = FALSE)
```

Arguments

cascades an object of class cascade containing node and cascade information. See [as_cascade_long](#) and [as_cascade_wide](#) for details.

trans_mod character, indicating the choice of model: "exponential" or "rayleigh".

n_edges integer, number of edges to infer.

lambda numeric, alpha for transmission model.

quiet logical, Should output on progress by suppressed.

Details

The algorithm is describe in detail in Gomez-Rodriguez et al. (2010). Additional information can be found on the netinf website (<http://snap.stanford.edu/netinf/>).

If higher performance is required and for very large data sets, a faster pure C++ implementation is available in the Stanford Network Analysis Project (SNAP). The software can be downloaded at <http://snap.stanford.edu/netinf/>.

Value

Returns the inferred diffusion network as an edgelist in an object of class `diffnet` and `data.frame`. The first column contains the sender, the second column the receiver node. The third column contains the improvement in fit from adding the edge that is represented by the row.

References

M. Gomez-Rodriguez, J. Leskovec, A. Krause. Inferring Networks of Diffusion and Influence. The 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2010.

Examples

```
# Data already in cascades format:
data(cascades)
out <- netinf(cascades, trans_mod = "exponential", n_edges = 5, lambda = 1)

# Starting with a dataframe
df <- simulate_rnd_cascades(10, n_nodes = 20)
cascades2 <- as_cascade_long(df, node_names = unique(df$node_name))
out <- netinf(cascades2, trans_mod = "exponential", n_edges = 5, lambda = 1)
```

NetworkInference

NetworkInference: Inferring latent diffusion networks

Description

This package provides an R implementation of the `netinf` algorithm created by Gomez Rodriguez, Leskovec, and Krause (2010). Given a set of events that spread between a set of nodes the algorithm infers the most likely stable diffusion network that is underlying the diffusion process.

Details

The package provides three groups of functions: 1) data preparation 2) estimation and 3) interpretation.

Data preparation

The core estimation function `netinf` requires an object of class `cascade` (see `as_cascade_long` and `as_cascade_wide`). Cascade data contains information on the potential nodes in the network as well as on event times for each node in each cascade.

Estimation

Diffusion networks are estimated using the `netinf` function. It produces a diffusion network in form of an edgelist (of class `data.frame`).

Interpretation and Visualization

Cascade data can be visualized with the `plot` method of the `cascade` class (`diffnet`, `plot.cascade`). Results of the estimation process can be visualized using the plotting method of the `diffnet` class.

Performance

If higher performance is required and for very large data sets, a faster pure C++ implementation is available in the Stanford Network Analysis Project (SNAP). The software can be downloaded at <http://snap.stanford.edu/netinf/>.

plot.cascade

Plot a cascade object

Description

Allows plotting of one or multiple, labeled or unlabeled cascades.

Usage

```
## S3 method for class 'cascade'
plot(x, label_nodes = TRUE, selection = NULL, ...)
```

Arguments

<code>x</code>	object of class <code>cascade</code> to be plotted.
<code>label_nodes</code>	logical, indicating if should the nodes in each cascade be labeled. If the cascades are very dense setting this to <code>FALSE</code> is recommended.
<code>selection</code>	a vector of cascade ids to plot.
<code>...</code>	additional arguments passed to plot.

Details

The function returns a `ggplot` plot object (class `gg`, `ggplot`) which can be modified like any other `ggplot`. See the `ggplot` documentation and the examples below for more details.

Value

A ggplot plot object.

Examples

```
data(cascades)
plot(cascades, selection = names(cascades$cascade_nodes)[1:5])
plot(cascades, label_nodes = FALSE, selection = sample(1:54, 20))

# Modify resulting ggplot object
library(ggplot2)
p <- plot(cascades, label_nodes = FALSE, selection = sample(1:54, 20))
## Add a title
p <- p + ggtitle('Your Title')
p
## Change Axis
p <- p + xlab("Your modified y axis label") #x and y labels are flipped here
p <- p + ylab("Your modified x axis label") #x and y labels are flipped here
p
```

plot.diffnet

Visualize netinf output

Description

Visualize the inferred diffusion network or the marginal gain in fit obtained by addition of each edge.

Usage

```
## S3 method for class 'diffnet'
plot(x, type = "network", ...)
```

Arguments

x	object of class diffnet to be plotted.
type	character, one of c("network", "improvement") indicating if the inferred diffusion network ("network") or the improvement for each edge should be visualized ("improvement").
...	additional arguments.

Details

If ‘type = improvement’ a ggplot object is returned. It can be modified like any other ggplot. See the ggplot documentation and the examples in [plot.cascade](#).

Value

A ggplot plot object if type = "improvement" otherwise an igraph plot.

Examples

```
## Not run:  
data(cascades)  
res <- netinf(cascades, n_edges = 6, lambda = 1)  
plot(res, type = "network")  
plot(res, type = "improvement")  
  
## End(Not run)
```

policies

US States Policy Adaption

Description

Adoption dates (years) for 187 policies across 50 US states. Compiled by Desmarais et al. (2015).

Usage

```
data(policies)
```

Format

The data is in a matrix format. Rows correspond to states (see rownames) columns to policies. Cell entries indicate the year a state adopted a policy. NA entries indicate states not having adopted a policy at all.

Source

<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/A1GIMB>

References

Desmarais, B. A., Harden, J. J., & Boehmke, F. J. (2015). Persistent Policy Pathways: Inferring Diffusion Networks in the American States. *American Political Science Review*, 109(02), 392-406.

simulate_cascades	<i>Simulate cascades from a diffusion network</i>
-------------------	---

Description

Simulate diffusion cascades based on the generative model underlying netinf and a diffusion network.

Usage

```
simulate_cascades(diffnet, nsim = 1, seed = NULL, max_time = Inf, lambda,
  beta, epsilon, model, partial_cascade = NULL, start_probabilities = NULL,
  nodes = NULL)
```

Arguments

diffnet	object of class diffnet.
nsim	integer, number of cascades to simulate.
seed	integer, seed for random number generator.
max_time	numeric, the maximum time after which observations are censored
lambda	numeric, parameter for diffusion time distribution.
beta	numeric, weight for in-network diffusion
epsilon	numeric, weight for out of network diffusion
model	character, diffusion model to use. One of c("exponential", "rayleigh").
partial_cascade	object of type cascade, containing one partial cascades for which further development should be simulated.
start_probabilities	a vector of probabilities for each node in diffnet, to be the node with the first event. If NULL a node is drawn from a uniform distribution over all nodes.
nodes	vector of node names, optional, if nodes should be included in simulation that do not occur in diffnet

Value

A data frame with three columns. Containing 1) The names of the nodes ("node_name") that experience an event in each cascade, 2) the event time ("event_time") of the corresponding node, 3) the cascade identifier "cascade_id".

Examples

```
data(cascades)
out <- netinf(cascades, trans_mod = "exponential", n_edges = 5, lambda = 1)
simulated_cascades <- simulate_cascades(out, nsim = 10, lambda = 1,
                                       beta = 0.5, epsilon = 10^-9,
                                       model = "exponential")

# Simulation from partial cascade
```

simulate_rnd_cascades *Simulate a set of random cascades*

Description

Simulate random cascades, for testing and demonstration purposes. No actual diffusion model is underlying these cascades.

Usage

```
simulate_rnd_cascades(n_cascades, n_nodes)
```

Arguments

n_cascades	Number of cascades to generate.
n_nodes	Number of nodes in the system.

Value

A data frame containing (in order of columns) node ids, event time and cascade identifier.

Examples

```
df <- simulate_rnd_cascades(10, n_nodes = 20)
head(df)
```

subset_cascade	<i>Select a subset of cascades from cascade object</i>
----------------	--

Description

Select a subset of cascades from cascade object

Usage

```
subset_cascade(cascade, selection)
```

Arguments

cascade	cascade, object to select from
selection	character or integer, vector of cascade_ids to select

Value

An object of class cascade containing just the selected cascades

Examples

```
data(policies)
cascades <- as_cascade_wide(policies)
cascade_names <- names(cascades$cascade_times)
subset_cascade(cascades, selection = cascade_names[1:10])
```

subset_cascade_time	<i>Subset a cascade object in time</i>
---------------------	--

Description

Remove each all events occurring outside the desired subset for each cascade in a cascade object.

Usage

```
subset_cascade_time(cascade, start_time, end_time, drop = TRUE)
```

Arguments

cascade	cascade, object to subset.
start_time	numeric, start time of the subset.
end_time	numeric, end time of the subset.
drop	logical, should empty sub-cascades be dropped?

Value

An object of class `cascade`, where only events are included that have times `start_time <= t < end_time`.

Examples

```
data(cascades)
sub_cascades <- subset_cascade_time(cascades, 10, 20, drop=TRUE)
```

<code>summary.cascade</code>	<i>Summarize a cascade object</i>
------------------------------	-----------------------------------

Description

Generates summary statistics for single cascades and across cascades in a collection, contained in a `cascades` object.

Usage

```
## S3 method for class 'cascade'
summary(object, quiet = FALSE, ...)
```

Arguments

<code>object</code>	object of class <code>cascade</code> to be summarized.
<code>quiet</code>	logical, if <code>FALSE</code> summary stats are printed to std out.
<code>...</code>	Additional arguments passed to <code>summary</code> .

Value

Prints cascade summary information to the screen (if `quiet = FALSE`). '# cascades' is the number of cascades in the object, '# nodes' is the number of nodes in the system (nodes that can theoretically experience an event), '# nodes in cascades' is the number of unique nodes of the system that experienced an event and '# possible edges' is the number of edges that are possible given the cascade data (see [count_possible_edges](#) for details.).

Additional summaries for each cascade are returned invisibly. `cascade`, `length` (length of the cascade as an integer of how many nodes experienced and event) and `n_ties` (number of tied event times per cascade).

Examples

```
data(cascades)
summary(cascades)
```

`validation`*Validation output from netinf source.*

Description

Contains output from original netinf C++ implementation, executed on [cascades](#). For testing purposes.

Usage

```
data(validation)
```

Format

An object of class `data.frame` with 6 columns, containing:

origin_node Origin of diffusion edge.

destination_node Destination node of diffusion edge.

volume ??

marginal_gain Marginal gain from edge.

median_time_difference Median time between events in origin and destination

mean_time_difference Mean time between events in origin and destination

Source

Output from netinf example program (<https://github.com/snap-stanford/snap/tree/master/examples/netinf>).

Index

*Topic **datasets**

- cascades, [6](#)
- policies, [12](#)
- validation, [17](#)

- as.data.frame.cascade, [2](#)
- as.matrix.cascade, [3](#)
- as_cascade_long, [4](#), [8](#), [10](#)
- as_cascade_wide, [5](#), [8](#), [10](#)

- cascades, [6](#), [17](#)
- count_possible_edges, [6](#), [16](#)

- data.frame, [2](#), [4](#), [5](#), [9](#), [10](#)

- is.cascade, [7](#)
- is.diffnet, [8](#)

- matrix, [3](#), [5](#)

- netinf, [8](#), [8](#), [10](#)
- NetworkInference, [9](#)

- plot.cascade, [10](#), [10](#), [11](#)
- plot.diffnet, [11](#)
- policies, [12](#)

- simulate_cascades, [13](#)
- simulate_rnd_cascades, [14](#)
- subset_cascade, [15](#)
- subset_cascade_time, [15](#)
- summary.cascade, [16](#)

- validation, [17](#)