Package 'bbnet'

May 18, 2024

Type Package

Title Create Simple Predictive Models on Bayesian Belief Networks

Version 1.0.1

Maintainer Victoria Dominguez Almela <vda1r22@soton.ac.uk>

Description A system to build, visualise and evaluate Bayesian belief networks. The methods are described in Stafford et al. (2015) <doi:10.12688/f1000research.5981.1>.

License GPL (>= 2)

URL https://github.com/vda1r22/bbnet

BugReports https://github.com/vda1r22/bbnet/issues

Depends R (>= 3.5.0), dplyr, ggplot2, grid, igraph, tibble

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Victoria Dominguez Almela [cre, aut]
(<https://orcid.org/0000-0003-4877-5967>),
Richard Stafford [cph, aut] (<https://orcid.org/0000-0002-1964-5787>)

Repository CRAN

Date/Publication 2024-05-18 13:20:03 UTC

R topics documented:

`bbn.network.diagram` *Create Network Diagram from Bayesian Belief Network Data*

Description

`bbn.network.diagram()` generates a network diagram from a specified Bayesian Belief Network (`BBN`), allowing for the visualization of the relationships and interactions between `nodes`.

Usage

```
bbn.network.diagram(
  bbn.network,
  font.size = 0.7,
  arrow.size = 4,
  arrange = layout_on_sphere
)
```

Arguments

`bbn.network`	A dataframe, with a first column called `id` that consists of an `s` and a 2 digit number relating to the `node` number. The second column called `node.type` is an `integer` value from 1-4. This sets the colour of the `node` in the network (sticking to a maximum of four colours). The third column is the same as the first column in the standard `BBN` interaction matrix or dataframe, other than it is titled `node.name`. It is important to use these column names (including capitals and dot notation). The remainder of the columns are exactly as the standard `BBN` interaction matrix or dataframe.
`font.size`	Changes the font in the figure produced. Default = 0.7. The value here is a multiplier of the default font size used in the `igraph` package and does not correspond to the `font.size` argument in `bbn.timeseries`.
`arrow.size`	Changes the size of the arrows. Default = 4. Note, sizes do vary based on interaction strength, so this is a multiplier for visualisation purposes. Negative interactions are shown by red arrows, and positive interactions by black arrows.

arrange — this describes how the final diagram looks. Default is layout_on_sphere but layout_on_grid provides the same layout as in bbn.visualise and ensures nodes are structured in the order specified in the network. Other layouts, including layout_on_sphere are more randomly determined, and better/clearer diagrams may occur if you run these multiple times. Other options are from the igraph package: layout.sphere layout.circle layout.random layout.fruchterman.reingold

Details

The diagram is created using edge and node data derived from the BBN, with edges representing interactions (positive or negative) between nodes.

bbn.network.diagram() visualises all nodes and interactions in a network, in a similar manner to bbn.visualise, other than this is the full network.

Value

A plot of the network diagram, illustrating the interactions (both positive and negative) between nodes.

Examples

```
data(my_network)
bbn.network.diagram(bbn.network = my_network, font.size=0.7,
  arrow.size=4, arrange = layout_on_sphere)
```

---

bbn.predict *Bayesian Belief Network Prediction*

Description

bbn.predict performs predictions using a Bayesian Belief Network (BBN) model, accommodating multiple priors scenarios and allowing for bootstrapping to assess variability.

Usage

```
bbn.predict(
  bbn.model,
  ...,
  boot_max = 1,
  values = 1,
  figure = 1,
  font.size = 5
)
```

Arguments

bbn.model	A matrix or dataframe of interactions between different model nodes.
...	An X by 2 array of initial changes to the system under investigation. It requires at least 1 prior scenario (up to 12 priors). The first column should be a -4 to 4 (including 0) integer value for each node in the network with negative values indicating a decrease and positive values representing an increase. 0 represents no change. Note, names included here are included as outputs in tables and figures. Shortening these names can provide better figures.
boot_max	The number of bootstraps to perform. Suggested range for exploratory analysis 1-1000. For final analysis recommended size = 1000 - 10000 - note, this can take a long time to run. Default value is 1, running with no bootstrapping - suitable for exploration of data and error checking.
values	This provides a numeric output of posterior values and any confidence intervals. Default value 1. Set to 0 to hide this output.
figure	Sets the figure options. Default value 1. 0 = no figures produced. 1 = figure is saved in working directory as a PDF file (note, this is overwritten if the name is not changed, and no figure is produced if the existing PDF is open when the new one is generated). 2 = figure is produced in a graphics window. All figures are combined on a single plot where scenario 2 is below scenario 1 (i.e. scenarios work in columns then rows)
font.size	Font size for the plot labels. Defaults to 5.

Details

- Supports input of multiple priors through ellipsis().
- Allows bootstrapping with a specified number of maximum iterations to assess prediction variability.
- Generates plots for visual representation of the predictions.

Value

Plots of the (BBN) predictions and optionally prints the predicted values.

Examples

```
data(my_BBN, combined)
bbn.predict(bbn.model = my_BBN, priors1 = combined, boot_max=100, values=1, figure=1, font.size=5)
```

bbn.sensitivity *Sensitivity Analysis for Bayesian Belief Network Models*

Description

bbn.sensitivity() conducts a sensitivity analysis on a Bayesian Belief Network (BBN) model. It evaluates the impact of varying key node on the network's outcomes using bootstrapping. The analysis helps identify which node significantly influence the network, providing insights into the robustness and dependency of the network's structure.

Usage

```
bbn.sensitivity(bbn.model, boot_max = 1000, ...)
```

Arguments

bbn.model	a matrix or dataframe of interactions between different model nodes. One or more nodes (recommended no more than 3) which would be the main outcomes of interest in the model. The spelling of these nodes needs to be identical (including capital letters) to that in the matrix or dataframe file. (note, you should include spaces if these are in your matrix or dataframe file, rather than the dot notation used once imported into R).
boot_max	The number of bootstraps to perform. Suggested range for exploratory analysis 100-1000. For final analysis recommended size = 1000 - 10000 - note, this can take a long time to run. Default value is 1000.
...	Key nodes for sensitivity analysis. The function is designed to handle up to three key nodes, beyond which it recommends limiting the analysis for clarity and efficiency.

Value

The function outputs a plot showing the nodes most influential to the network's outcomes, alongside a table ranking these variables by their impact. The analysis highlights how changes in the key nodes can affect the network, offering valuable insights for model refinement and decision-making.

Examples

```
data(my_BBN)
bbn.sensitivity(bbn.model = my_BBN, boot_max = 100, 'Limpet', 'Green Algae')
```

bbn.timeseries *Time Series Prediction with Bayesian Belief Network*

Description

bbn.timeseries() performs time series predictions using a Bayesian Belief Network (BBN) model based on a single prior scenario. It generates figures illustrating how parameters change over time for all or selected nodes.

Usage

```
bbn.timeseries(bbn.model, priors1, timesteps = 5, disturbance = 1)
```

Arguments

bbn.model	A matrix or dataframe of interactions between different model nodes.
priors1	An X by 2 array of initial changes to the system under investigation. The first column should be a -4 to 4 (including 0) integer value for each node in the network with negative values indicating a decrease and positive values representing an increase. 0 represents no change.
timesteps	This is the number of timesteps the model performs. Default = 5. Note, timesteps are arbitrary and non-linear. However, something occurring in timestep 2, should occur before timestep 3.
disturbance	Default = 1. 1 creates a prolonged or press disturbance as per bbn.predict. Essentially prior values for each manipulated node are at least maintained (if not increased through reinforcement in the model) over all timesteps. 2 shows a brief pulse disturbance, which can be useful to visualise changes as peaks and troughs in increase and decrease of nodes can propagate through the network.

Value

Plots for each node showing the predicted change over time.

Examples

```
data(my_BBN, combined)
bbn.timeseries(bbn.model = my_BBN, priors1 = combined, timesteps=6, disturbance=1)
```

bbn.visualise *Visualise Bayesian Belief Network Time Series Predictions*

Description

bbn.visualise() visualises the outcomes of a Bayesian Belief Network (BBN) model over time, given a single prior scenario. It highlights the changes in network parameters across specified timesteps and visualises the strength and direction of interactions among nodes based on the specified disturbance and threshold parameters.

Usage

```
bbn.visualise(
  bbn.model,
  priors1,
  timesteps = 5,
  disturbance = 1,
  threshold = 0.2,
  font.size = 0.7,
  arrow.size = 4
)
```

Arguments

bbn.model	A matrix or dataframe of interactions between different model nodes.
priors1	An X by 2 array of initial changes to the system under investigation. The first column should be a -4 to 4 (including 0) integer value for each node in the network with negative values indicating a decrease and positive values representing an increase. 0 represents no change.
timesteps	This is the number of timesteps the model performs. Default = 5. Note, timesteps are arbitrary and non-linear. However, something occurring in timestep 2, should occur before timestep 3.
disturbance	Default = 1. 1 creates a prolonged or press disturbance as per bbn.predict Essentially prior values for each manipulated node are at least maintained (if not increased through reinforcement in the model) over all timesteps. 2 shows a brief pulse disturbance, which can be useful to visualise changes as peaks and troughs in increase and decrease of nodes can propagate through the network.
threshold	Nodes which deviate from 0 by more than this threshold value will display interactions with other nodes. Default = 0.2. Values in these visualisation functions don't directly correspond to those in bbn.predict. This value can be tweaked from 0 to 4 to create the most useful visualisations.
font.size	Changes the font in the figure produced. Default = 0.7. The value here is a multiplier of the default font size used in the igraph package and does not correspond to the font.size argument in the bbn.timeseries.
arrow.size	Changes the size of the arrows. Default = 4. Note, sizes do vary based on interaction strength, so this is a multiplier for visualisation purposes.

Value

A plot of the BBN, illustrating the dynamic interactions between nodes over the specified timesteps.

Examples

```
data(my_BBN, combined)
bbn.visualise(bbn.model = my_BBN, priors1 = combined,
  timesteps=6, disturbance=1, threshold=0.2, font.size=0.7, arrow.size=4)
```

---

combined *Combined Treatment Data*

Description

This dataset represents the numerical changes in species populations on a rocky shore ecosystem due to the combined treatment of removing dogwhelks and adding periwinkles. It reflects the complex interactions and potential synergistic effects of multiple ecological interventions.

Format

A data frame with 9 rows and 2 columns:

Increase integer

Node Variable names

Source

<https://doi.org/10.1016/j.ocecoaman.2015.04.013>

---

dogwhelk *Dogwhelk Removal Data*

Description

This dataset represents the numerical changes in species populations on a rocky shore ecosystem due to the removal of dogwhelks. It provides insights into the potential ecological impacts of removing a predatory species.

Format

A data frame with 9 rows and 2 columns:

Increase integer

Node Variable names

Source

<https://doi.org/10.1016/j.ocecoaman.2015.04.013>

isEmpty *Check if an Object is Empty*

Description

This function determines whether the provided object is empty.

Usage

```
isEmpty(x)
```

Arguments

x	The object to check for emptiness.

Details

`isEmpty()` checks if the given object, `x`, has a length of 0, indicating that it is empty. It can be used with various types of objects in R, including vectors, lists, and data frames.

Value

A logical value: `TRUE` if the object is empty, `FALSE` otherwise.

Examples

```
# Check an empty vector
isEmpty(c())

# Check a non-empty vector
isEmpty(c(1, 2, 3))

# Check an empty list
isEmpty(list())

# Check a non-empty list
isEmpty(list(a = 1, b = 2))

# Check an empty data frame
isEmpty(data.frame())

# Check a non-empty data frame
isEmpty(mtcars)
```

MPANetwork *Simple model of MPA, ecological components and management*

Description

This dataset represents an interaction model of marine protected area and ecological components This is an example dataset loosely based on Lundy Island MCZ.

Format

A data frame with 11 rows and 12 columns:

id Variable names

Lobster.fishery integer

Finfish.fishery integer

Fish.density integer

Seals integer

Lobster.Recruitment integer

Divers integer

Spiny.lobster integer

Lobster integer

Snails integer

Algae integer

Revenue integer

Source

<unpublished work by Rick Stafford>

multiplot *Multiplot function*

Description

This function allows for the arrangement and display of multiple `ggplot2` plots on a single graphics page.

Usage

```
multiplot(..., plotlist = NULL, file, cols = 1, layout = NULL)
```

Arguments

...	One or more ggplot2 objects to be plotted.
plotlist	An optional list of ggplot2 objects. This parameter can be used in conjunction with or instead of the direct plot arguments.
file	A path to save the output file.
cols	Specifies the number of columns in the grid layout if layout is not provided. Defaults to 1.
layout	An optional matrix specifying the layout of plots. Overrides cols if provided.

Details

multiplot() can take any number of plot objects as arguments, or if it can take a list of plot objects passed to plotlist.

multiplot() is built under CC0 licence from:

http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/

ggplot2 objects can be passed in ..., or to plotlist (as a list of ggplot2 objects)

Details:

- cols: Number of columns in layout.
- layout: A matrix specifying the layout.If present, cols is ignored.

If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE), then plot 1 will go in the upper left, 2 will go in the upper right, and 3 will go all the way across the bottom.

Value

plot

Examples

```
# Load necessary library
library(ggplot2)

# Create example ggplot objects
plot1 <- ggplot(mtcars, aes(x=mpg, y=wt)) + geom_point()
plot2 <- ggplot(mtcars, aes(x=mpg, y=cyl)) + geom_point()
plot3 <- ggplot(mtcars, aes(x=gear, y=wt)) + geom_point()

# Plot all three plots in a single row
multiplot(plot1, plot2, plot3, cols=3)

# Plot using a custom layout
layout_matrix <- matrix(c(1,2,3,3), nrow=2, byrow=TRUE)

multiplot(plotlist=list(plot1, plot2, plot3), layout=layout_matrix)
```

my_BBN *Rocky Shore simple food web data*

Description

This dataset represents a simplified food web of a rocky shore ecosystem, focusing on the interactions between various species. The data was used to study the effects of various ecological interventions and their effects, as described in the associated paper.

Format

A data frame with 9 rows and 10 columns:

X Row names, representing various species

Dogwhelk integer

Topshell integer

Limpet integer

Periwinkle integer

Barnacle integer

Green.Algae integer

Biofilm integer

Corline.algae integer

Fucoid.Algae integer

Source

<https://doi.org/10.1016/j.ocecoaman.2015.04.013>

my_network *Rocky Shore complex food web data*

Description

In this file, the first column is called id and consists of an s and a 2 digit number relating to the node number. The second column is called node.type and is an integer value from 1-4. This sets the colour of the node in the network (sticking to a maximum of four colours). Here, predators, grazers, filter feeders and algae are colour coded separately it would be fine to change the colours, for example to ensure algae were green. The third column is the same as the first column in the standard BBN interaction csv, other than it is titled node.name. It is important to use these column names (including capitals and dot notation). The remainder of the columns are exactly as the standard my_BBN data file.

Format

A data frame with 9 rows and 12 columns:

id Variable names

node.type integer

node.name Variable names

Dogwhelk integer

Topshell integer

Limpet integer

Periwinkle integer

Barnacle integer

Green.Algae integer

Biofilm integer

Corline.algae integer

Fucoid.Algae integer

Source

<https://doi.org/10.1016/j.ocecoaman.2015.04.013>

NoPotting *Dataset represents banning potting (for crabs / lobsters etc) in a Marine Protected Area Data presents insights into how management measures may affect ecological communities in MPAs*

Description

Dataset represents banning potting (for crabs / lobsters etc) in a Marine Protected Area Data presents insights into how management measures may affect ecological communities in MPAs

Format

A data frame with 11 rows and 2 columns:

Increase integer

Node Variable names

Source

<unpublished work by Rick Stafford>

NoTake *Dataset represents banning all fishing in a Marine Protected Area*

Description

Data presents insights into how management measures may affect ecological communities in MPAs

Format

A data frame with 11 rows and 2 columns:

Increase integer

Node Variable names

Source

<unpublished work by Rick Stafford>

winkle *Winkle Addition Data*

Description

This dataset represents the numerical changes in species populations on a rocky shore ecosystem due to the addition of periwinkles. It captures the direct interventions and expected ecological shifts as modeled in the study.

Format

A data frame with 9 rows and 2 columns:

Increase integer

Node Variable names

Source

<https://doi.org/10.1016/j.ocecoaman.2015.04.013>

Index