shinyBN: An online application for interactive Bayesian network inference and visualization

Manual catalog:

【Input section】

Requirements of input files:

1. **R object**
   It must be the output of package *bnlearn* and in two class:
   - bn: Network structure for visualization only
   - bn.fit: Network with parameters for inference.
2. **Individual Data (.csv)**
   The data is an **N×M** matrix with discrete data, where **N** is the number of observables and **M** is the number of the features (nodes).

	X_1	X_2	X_3	…
ID1	High	1	0	…
ID2	Low	0	1	…
	…			

3. **Network Structure (Excel)**
   This Excel file contains 2 sheets:
   - Nodes: This sheet contains 9 columns, of which the required is in **red**.

Collumn	**Meaning**	**Value**
id	The id of the nodes have to be unique.	String.
label	The label of nodes to display.	String.
x	This gives a node an initial x position.	Number.
y	This gives a node an initial y position.	Number.
color	Color for the nodes.	String. Such as ’#97C2FC’, ’rgba(120,32,14,1)’ or ’red’.
shape	Shape for the nodes.	"ellipse", "circle", "box", "database".
font.size	The size is used to determine the size of node shapes.	Number.
font.color	Color of the label.	String. Such as ’#97C2FC’, ’rgba(120,32,14,1)’ or ’red’.
group	The group of nodes for fast render.	String.

   - Edges: This sheet contains 6 columns, of which the required is in **red**.

Collumn	**Meaning**	**Value**
from	The id of the nodes have to be unique.	String.
to	The label of nodes to display.	String.
color	Color for the nodes.	String. Such as ’#97C2FC’, ’rgba(120,32,14,1)’ or ’red’.
width	The width of edges.	Number.
linetype	The type of the edges.	String. "solid" or "dashed"
group	The group of edges for fast render.	String.

【Network Construction for individual data】

1. Structure Learning:

Package: *bnlearn*

a) Constraint-Based Algorithm

i. Grow-Shrink
    gs(data, whitelist, blacklist, test, alpha)
ii. Incremental Association
    iamb(data, whitelist, blacklist, test, alpha)
iii. Fast Incremental Association
    fast.iamb(data, whitelist, blacklist, test, alpha)
iv. Interleaved Incremental Association
    inter.iamb(data, whitelist, blacklist, test, alpha)
v. Max-Min Parents and Children
    mmpc(data, whitelist, blacklist, test, alpha)
vi. Semi-Interleaved HITON-PC
    si.hiton.pc(data, whitelist, blacklist, test, alpha)

Options:

data	A data frame containing the variables in the model.
whitelist	A data frame with two columns (optionally labeled "from" and "to"), containing a set of arcs not to be included in the graph.
blacklist	A data frame with two columns (optionally labeled "from" and "to"), containing a set of arcs not to be included in the graph.
test	A character string, the label of the conditional independence test to be used in the algorithm. '*mutual information*'(Default), 'shrinkage estimator for the mutual information', 'Pearson's X^2'.
alpha	A numeric value, the target nominal type I error rate. Default 0.05.

b) Score-Based Algorithm

i. Hill-climbing
    hc(data, whitelist, blacklist, score, restart, perturb)
ii. Tabu search
    tabu(data, whitelist, blacklist, score, tabu)

Options:

data	A data frame containing the variables in the model.
whitelist	A data frame with two columns (optionally labeled "from" and "to"), containing a set of arcs not to be included in the graph.
blacklist	A data frame with two columns (optionally labeled "from" and "to"), containing a set of arcs not to be included in the graph.
score	A character string, the label of the network score to be used in the algorithm. 'Bayesian Information Criterion score' (Default), 'Akaike Information Criterion score','Multinomial log-

	likelihood score',‘Bayesian Dirichlet equivalent score','Bayesian Dirichlet sparse score', 'Modified Bayesian Dirichlet equivalent score', ‘Locally averaged Bayesian Dirichlet score', 'K2 score'
restart	An integer, the number of random restarts. Default 0.
perturb	An integer, the number of attempts to randomly insert/ remove/reverse an arc on every random restart. Default 1.
tabu	A positive integer number, the length of the tabu list used in the tabu function. Default 10.

c) Hybrid Algorithm

i. Max-Min Hill Climbing

mmhc(data, whitelist, blacklist, restrict, maximize, restrict.args, maximize.args)

ii. 2-phase Restricted Maximization

rsmax2(data, whitelist, blacklist, restrict, maximize, restrict.args, maximize.args)

Options:

data	A data frame containing the variables in the model.
whitelist	A data frame with two columns (optionally labeled "from" and "to"), containing a set of arcs not to be included in the graph.
blacklist	A data frame with two columns (optionally labeled "from" and "to"), containing a set of arcs not to be included in the graph.
restrict	A character string, the constraint-based or local search algorithm to be used in the “restrict” phase. Default ‘gs'
maximize	A character string, the score-based algorithm to be used in the “maximize” phase. Default ‘hc
restrict.args	A list of arguments to be passed to the algorithm specified by restrict, such as test or alpha.
maximize.args	A list of arguments to be passed to the algorithm specified by maximize, such as restart for hill-climbing or tabu for tabu search.

d) Bootstrap

bn.boot(data, R, algorithm, algorithm.args)
averaged.network(strength, threshold)

Options:

data	A data frame containing the variables in the model.
algorithm	A character string, the learning algorithm to be applied to the bootstrap replicates. Possible values are gs, iamb, fast.iamb, inter.iamb, mmpc, hc, tabu, mmhc and rsmax2.
algorithm.args	A list of extra arguments to be passed to the learning algorithm.
strength	An object of class bn.strength (The output of bn.boot)
threshold	A numeric value, the minimum strength required for an arc to

be included in the averaged network. Default 0.85 .

2. Parameter Learning:

Package: *bnlearn*

Description

Fit the parameters of a Bayesian network conditional on its structure.

bn.fit(x, data, method)

Options:

x	An object of class bn.(The output of structure learning).
data	A data frame containing the variables in the model.
method	A character string, either 'Maximum Likelihood parameter estimation' (Default) or 'Bayesian parameter estimation'.

【Network Visualization】

Package: *visNetwork*

Description

Network visualization using vis.js library.

```
visNetwork() %>%
    visNodes%>%
    visEdges%>%
    visLayout%>%
    visLegend
```

Options:

Nodes	color	Self-defined(lightblue, red…), SCI-style(NPG, JAMA, Lancent…), Pic-style(Upload a picture to extract the contrast colors)
	shape	'ellipse', 'circle', 'database', 'box'
	label size	Number. Size of the label text and nodes.
	label color	Color of the label text.
Edges	color	String. 'gray', 'red', 'orange', 'yellow'…
	line type	'solid', 'dashed'
	width	Number. The width of the edge. Can be defined by self or corresponding to arc strength.
Layout	Layout	Valid when node positions are not specified. 'Layer'(Default), 'Circle', 'Star', 'Tree' ,'Grid'
Legend	key size	Number. The size of the legend key.
	position	The position of the legend. 'right'(Default), 'left'

【Network Inference】

Package: *gRain*

1. Single Prediction

Steps:

1) Select the Predictors

setEvidence(object, nodes, states)

Options:

object	A "grain" object (The output of function *as.grain* in *bnlearn* that convert bn.fit objects to grain objects)
nodes	A vector of nodes; those nodes for which the (conditional) distribution is requested.
states	A vector of states (of the nodes given by ’nodes’)

2) Choose the Outcome

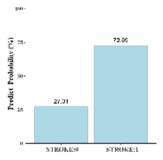
querygrain(object, nodes, type)

Options:

object	A "grain" object (The output of function *setEvidence*)
nodes	A vector of nodes; those nodes for which the (conditional) distribution is requested.
type	"marginal"(Default) gives the marginal distribution for each node in nodes; "joint" gives the joint distribution for nodes

3) Barplot Output

ggplot:



2. Validation Set (By batch)

Steps:

1) Upload validation set

Requirements:

This file should have the same structure as Individual Data (.csv). If this file contains the label of outcome, it can be used as a validation set and perform batch inference, ROC plot and DCA plot. If not, it can be used to batch inference only. When there are missing predictors, *shinyBN* would perform prediction based on non-missing predictor. Cases with missing outcome would be deleted for ROC plot and DCA plot.

2) Choose the Outcome

3) Result Download

i. Batch inference.

A csv file have the same structure as uploaded validation set but have more columns about the predicted results.

ii. ROC plot

Package: *pROC*

roc(response, predictor, smooth, ci=T, of='auc')

Options:

response	A factor, numeric or character vector of responses, typically encoded with 0 (controls) and 1 (cases).
predictor	A numeric or ordered vector of the same length than response, containing the predicted value of each observation.
smooth	If TRUE, the ROC curve is passed to smooth to be smoothed. Default FALSE.

plot.roc(x,legacy.axes=T, col, lty, lwd, print.thres, print.thres.col, print.thres.cex, print.auc,print.auc.col,print.auc.cex, print.auc.ci, grid, grid.col, grid.lty, grid.lwd, auc.polygon, auc.polygon.col)

Options:

x	A roc object from the roc function
col, lty, lwd	The color, line type and line width for the ROC curve
print.auc	boolean. Should the numeric value of AUC be printed on the plot? Default TRUE.
print.auc.col	The color for the printing of the AUC. Default 'black'.
print.auc.cex	The character expansion factor for the printing of the AUC.
print.auc.ci	boolean. Should the confidence interval of AUC be printed on the plot? Default TRUE.
print.thres	Should a selected set of thresholds be displayed on the ROC curve? Default TRUE.
print.thres.col	The color for the printing of the thresholds. Default 'black'.
print.thres.cex	The character expansion factor for the printing of the thresholds. Default 1.5.
auc.polygon	boolean. Whether or not to display the area as a polygon. Default FALSE.
auc.polygon.col	The color for the AUC polygon. Default 'lightblue'.
grid	Boolean. Should a background grid be added to the plot? Default FALSE.
grid.col	The color of the lines of the grid. Default 'gray'.
grid.lty	The line type of the lines of the grid. Default 'solid'.
grid.lwd	The line width of the lines of the grid. Default 0.5.

iii. DCA plot

Package: *rmda*

decision_curve(outcome, predictors, data, fitted.risk=T, family=

binomial(link = "logit"))

Options:

outcome	A factor, numeric or character vector of responses, typically encoded with 0 (controls) and 1 (cases).
predictor	A numeric or ordered vector of the same length than response, containing the predicted value of each observation.
data	data.frame containing outcome and predictors. Missing data on any of predictors will cause entire observation to be removed.

plot_decision_curve(model, curve.names, standardize, col, lty, lwd, xlim, ylim, xlab, ylab, legend.position, cost.benefit.axis=F, axe=T)

Options:

model	‘decision_curve’ object to plot. Assumes output from function ’decision_curve’
curve.names	Vector of names to use when plotting legends. Default ‘DCA model’
standardize	Logical. Indicating whether to use standardized net benefit(NB/disease prevalence) or not. Default TRUE.
col, lty, lwd	Vector of color, linetypes, linewidths to be used in plotting corresponding to the ’predictors’ given.
xlim, ylim	Vector giving c(min, max) of x-axis and y-axis. Defaults to ‘0,1’
xlab, ylab	Label of main x-axis and y-axis.
legend.position	Character vector giving position of legend. Options are "topright", "right", "bottomright", "bottom", "bottomleft", "left", "topleft", "top", or "none" (Default).

【Source code & Help】

1. Source code & Help

shinyBN is an open source project, and the source code and its manual is freely available at https://github.com/JiajinChen/shinyBN.

2. Contact us

If you have any problem or other inquiries you can also email us at fengchen@njmu.edu.cn or ywei@njmu.edu.cn .