Package ‘netropy’

October 13, 2022

Title Statistical Entropy Analysis of Network Data

Version 0.1.0

Description Statistical entropy analysis of network data as introduced by Frank and Shafie (2016) <doi:10.1177/0759106315615511>, and in a forthcoming book by Nowicki, Shafie and Frank (2022).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports ggraph, ggplot2, igraph

RoxygenNote 7.1.2

Language en-US

Depends R (>= 3.6)

Suggests testthat (>= 3.0.0), rmarkdown, knitr

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Termeh Shafie [aut, cre]

Maintainer Termeh Shafie <termeh@schochastics.net>

Repository CRAN

Date/Publication 2022-02-02 08:20:02 UTC

R topics documented:

`assoc_graph` *Association Graphs*

Description

Draws association graphs (graphical models) based on joint entropy values to detect and visualize different dependence structures among the variables in the dataframe.

Usage

```
assoc_graph(dat, cutoff = 0)
```

Arguments

`dat` dataframe with rows as observations and columns as variables. Variables must all be observed or transformed categorical with finite range spaces.

`cutoff` the cutoff point for the edges to be drawn based on joint entropies. Default is 0 and draws all edges.

Details

Draws association graphs based on given thresholds of joint entropy values between pairs of variables represented as nodes. Thickness of edges between pairs of nodes/variables indicates the strength of dependence between them. Isolated nodes are completely independent and paths through certain nodes/variables indicate conditional dependencies.

Value

A ggraph object with nodes representing all variables in `dat` and edges representing (the strength of) associations between them based on joint entropies.

Author(s)

Termeh Shafie

References

Frank, O., & Shafie, T. (2016). Multivariate entropy analysis of network data. *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique*, 129(1), 45-63.

Nowicki, K., Shafie, T., & Frank, O. (Forthcoming 2022). *Statistical Entropy Analysis of Network Data*.

See Also

`joint_entropy`

Examples

```
library(ggraph)
# use internal data set
data(lawdata)
df.att <- lawdata[[4]]

# three steps of data editing:
# 1. categorize variables 'years' and 'age' based on
# approximately three equally size groups (values based on cdf)
# 2. make sure all outcomes start from the value 0 (optional)
# 3. remove variable 'senior' as it consists of only unique values (thus redundant)
df.att.ed <- data.frame(
   status   = df.att$status,
   gender   = df.att$gender,
   office   = df.att$office-1,
   years    = ifelse(df.att$years<=3,0,
              ifelse(df.att$years<=13,1,2)),
   age      = ifelse(df.att$age<=35,0,
                ifelse(df.att$age<=45,1,2)),
   practice = df.att$practice,
   lawschool= df.att$lawschool-1)

# association graph based on cutoff 0.15
assoc_graph(df.att.ed, 0.15)
```

`entropy_bivar` *Bivariate Entropy*

Description

Computes the bivariate entropies between all pairs of (discrete) variables in a multivariate data set.

Usage

```
entropy_bivar(dat)
```

Arguments

`dat`	dataframe with rows as observations and columns as variables. Variables must all be observed or transformed categorical with finite range spaces.

Details

The bivariate entropy $H(X,Y)$ of two discrete random variables X and Y can be used to check for functional relationships and stochastic independence between pairs of variables. The bivariate entropy is bounded according to

$$H(X) <= H(X,Y) <= H(X) + H(Y)$$

where $H(X)$ and $H(Y)$ are the univariate entropies.

Value

Upper triangular matrix giving bivariate entropies between pairs of variables given as rows and columns of the matrix. The univariate entropies are given in the diagonal.

Author(s)

Termeh Shafie

References

Frank, O., & Shafie, T. (2016). Multivariate entropy analysis of network data. *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique*, 129(1), 45-63.

Nowicki, K., Shafie, T., & Frank, O. (Forthcoming 2022). *Statistical Entropy Analysis of Network Data*.

See Also

`joint_entropy`, `entropy_trivar`, `redundancy`, `prediction_power`

Examples

```
# use internal data set
data(lawdata)
df.att <- lawdata[[4]]

# three steps of data editing:
# 1. categorize variables 'years' and 'age' based on
# approximately three equally size groups (values based on cdf)
# 2. make sure all outcomes start from the value 0 (optional)
# 3. remove variable 'senior' as it consists of only unique values (thus redundant)
df.att.ed <- data.frame(
   status   = df.att$status,
   gender   = df.att$gender,
   office   = df.att$office-1,
   years    = ifelse(df.att$years<=3,0,
              ifelse(df.att$years<=13,1,2)),
   age      = ifelse(df.att$age<=35,0,
                ifelse(df.att$age<=45,1,2)),
   practice = df.att$practice,
   lawschool= df.att$lawschool-1)

# calculate bivariate entropies
H.biv <- entropy_bivar(df.att.ed)
# univariate entropies are then given as
diag(H.biv)
```

`entropy_trivar` *Trivariate Entropy*

Description

Computes trivariate entropies of all triples of (discrete) variables in a multivariate data set.

Usage

```
entropy_trivar(dat)
```

Arguments

`dat`	dataframe with rows as observations and columns as variables. Variables must all be observed or transformed categorical with finite range spaces.

Details

Trivariate entropies can be used to check for functional relationships and stochastic independence between triples of variables. The trivariate entropy *H(X,Y,Z)* of three discrete random variables *X, Y* and *Z* is bounded according to

H(X,Y) <= H(X,Y,Z) <= H(X,Z) + H(Y,Z) - H(Z).

The increment between the trivariate entropy and its lower bound is equal to the expected conditional entropy.

Value

Dataframe with the first three columns representing possible triples of variables (`V1,V2,V3`) and the fourth column gives trivariate entropies `H(V1,V2,V3)`.

Author(s)

Termeh Shafie

References

Frank, O., & Shafie, T. (2016). Multivariate entropy analysis of network data. *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique*, 129(1), 45-63.

Nowicki, K., Shafie, T., & Frank, O. (Forthcoming 2022). *Statistical Entropy Analysis of Network Data*.

See Also

`entropy_bivar`, `prediction_power`

Examples

```
# use internal data set
data(lawdata)
df.att <- lawdata[[4]]

# three steps of data editing:
# 1. categorize variables 'years' and 'age' based on
# approximately three equally size groups (values based on cdf)
# 2. make sure all outcomes start from the value 0 (optional)
# 3. remove variable 'senior' as it consists of only unique values (thus redundant)
df.att.ed <- data.frame(
   status   = df.att$status,
   gender   = df.att$gender,
   office   = df.att$office-1,
   years    = ifelse(df.att$years<=3,0,
              ifelse(df.att$years<=13,1,2)),
   age      = ifelse(df.att$age<=35,0,
                ifelse(df.att$age<=45,1,2)),
   practice = df.att$practice,
   lawschool= df.att$lawschool-1)

# calculate trivariate entropies
H.triv <- entropy_trivar(df.att.ed)
```

---

get_dyad_var *Get Dyad Variables*

Description

Transforms vertex variables or observed directed/undirected ties into dyad variables.

Usage

```
get_dyad_var(var, type = "att")
```

Arguments

var	variable vector (actor attribute) or adjacency matrix (ties) to be transformed to a dyad variable.
type	either ’att’ for actor attribute (default) or ’tie’ for relations.

Details

Dyad variables are given as pairs of incident vertex variables or actor attributes. Here, unique pairs of original attribute values constitute the outcome space. Note that the actor attributes need to be categorical with finite range spaces. For example, binary attribute yields outcome space (0,0), (0,1), (1,0), (1,1) coded as (0),(1),(2),(3). Warning message is shown if actor attribute has too many unique outcomes as it will yield too many possible outcomes once converted in to a dyad variable.

For directed relations, pairs of indicators from the adjacency matrix constitute the four outcomes representing possible combinations of sending and receiving ties: (0,0), (0,1), (1,0), (1,1) coded as (0),(1),(2),(3).

For undirected relations, an indicator variable which is directly read from the adjacency matrix represents the dyadic variable.

Value

Dataframe with three columns: first two columns show the vertex pairs `u` and `v` where `u<v` , and the third column gives the value of the transformed dyadic variable `var`.

Author(s)

Termeh Shafie

References

Frank, O., & Shafie, T. (2016). Multivariate entropy analysis of network data. *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique*, 129(1), 45-63.

Nowicki, K., Shafie, T., & Frank, O. (Forthcoming 2022). *Statistical Entropy Analysis of Network Data*.

See Also

`get_triad_var`

Examples

```
# use internal data set
data(lawdata)
adj.advice <- lawdata[[1]]
adj.cowork <-lawdata[[3]]
df.att <- lawdata[[4]]

# three steps of data editing of attribute dataframe:
# 1. categorize variables 'years' and 'age' based on
# approximately three equally size groups (values based on cdf)
# 2. make sure all outcomes start from the value 0 (optional)
# 3. remove variable 'senior' as it consists of only unique values (thus redundant)
df.att.ed <- data.frame(
   status   = df.att$status,
   gender   = df.att$gender,
   office   = df.att$office-1,
   years    = ifelse(df.att$years<=3,0,
              ifelse(df.att$years<=13,1,2)),
   age      = ifelse(df.att$age<=35,0,
                ifelse(df.att$age<=45,1,2)),
   practice = df.att$practice,
    lawschool= df.att$lawschool-1)

# actor attribute converted to dyad variable
dyad.gend <- get_dyad_var(df.att.ed$gender, 'att')

# directed tie converted to dyad variable
dyad.adv <- get_dyad_var(adj.advice, 'tie')

# undirected tie converted to dyad variable
dyad.cwk <- get_dyad_var(adj.cowork, 'tie')
```

get_triad_var *Get Triad Variables*

Description

Transforms vertex variables or observed directed/undirected ties into triad variables.

Usage

```
get_triad_var(var, type = "att")
```

Arguments

`var`	variable vector (actor attribute) or adjacency matrix (ties) to be transformed to a triad variable.
`type`	either 'att' for actor attribute (default) or 'tie' for relations.

Details

For actor attributes, unique triples of original attribute values constitute the outcome space. Note that the actor attributes need to be categorical with finite range spaces. For example, binary attributes have 8 possible triadic outcomes (0,0,0),(1,0,0),(0,1,0),(1,1,0),(0,0,1),(1,0,1),(0,1,1),(1,1,1) which are coded 0-7. Warning message is shown if actor attribute has too many unique outcomes as it will yield too many possible outcomes once converted in to a triad variable.

For directed relations, a sequence of indicators of length 6 created from the adjacency matrix constitutes the 64 outcomes representing possible combinations of sending and receiving ties.

For undirected relations, triples of indicators are created from the adjacency matrix.

Value

Dataframe with four columns: first three columns show the vertex triad `u`, `v`, `w` , and the fourth column gives the value of the transformed triadic variable `var`.

Author(s)

Termeh Shafie

References

Frank, O., & Shafie, T. (2016). Multivariate entropy analysis of network data. *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique*, 129(1), 45-63.

Nowicki, K., Shafie, T., & Frank, O. (Forthcoming 2022). *Statistical Entropy Analysis of Network Data.*

See Also

get_dyad_var

Examples

```
# use internal data set
data(lawdata)
adj.advice <- lawdata[[1]]
adj.cowork <-lawdata[[3]]
df.att <- lawdata[[4]]

# three steps of data editing:
# 1. categorize variables 'years' and 'age' based on
# approximately three equally size groups (values based on cdf)
# 2. make sure all outcomes start from the value 0 (optional)
# 3. remove variable 'senior' as it consists of only unique values (thus redundant)
df.att.ed <- data.frame(
   status   = df.att$status,
   gender   = df.att$gender,
   office   = df.att$office-1,
   years    = ifelse(df.att$years<=3,0,
              ifelse(df.att$years<=13,1,2)),
   age      = ifelse(df.att$age<=35,0,
                ifelse(df.att$age<=45,1,2)),
   practice = df.att$practice,
   lawschool= df.att$lawschool-1)

# actor attribute converted to triad variable
triad.gend <- get_triad_var(df.att.ed$gender, 'att')

# directed tie converted to triad variable
triad.adv <- get_triad_var(adj.advice, type = 'tie')

# undirected tie converted to triad variable
triad.cwk <- get_triad_var(adj.cowork, type = 'tie')
```

`joint_entropy` *Joint Entropy*

Description

Computes the joint entropies between all pairs of (discrete) variables in a multivariate data set.

Usage

```
joint_entropy(dat, dec = 3)
```

Arguments

`dat`	dataframe with rows as observations and columns as variables. Variables must all be observed or transformed categorical with finite range spaces.
`dec`	the precision given in number of decimals for which the frequency distribution of unique entropy values is created. Default is 3.

Details

The joint entropy $J(X,Y)$ of discrete variables X and Y is a measure of dependence or association between them, defined as

$$J(X,Y) = H(X) + H(Y) - H(X,Y).$$

Two variables are independent if their joint entropy, i.e. their mutual information, is equal to zero. The frequency distributions can be used to decide upon convenient thresholds for constructing association graphs.

Value

List with

`matrix`	an upper triangular joint entropy matrix (univariate entropies in the diagonal).
`freq`	a dataframe giving the frequency distributions of unique joint entropy values.

Author(s)

Termeh Shafie

References

Frank, O., & Shafie, T. (2016). Multivariate entropy analysis of network data. *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique*, 129(1), 45-63.

Nowicki, K., Shafie, T., & Frank, O. (Forthcoming 2022). *Statistical Entropy Analysis of Network Data*.

See Also

assoc_graph, entropy_bivar

Examples

```
# use internal data set
data(lawdata)
df.att <- lawdata[[4]]

# three steps of data editing:
# 1. categorize variables 'years' and 'age' based on
# approximately three equally size groups (values based on cdf)
# 2. make sure all outcomes start from the value 0 (optional)
# 3. remove variable 'senior' as it consists of only unique values (thus redundant)
df.att.ed <- data.frame(
   status   = df.att$status,
   gender   = df.att$gender,
   office   = df.att$office-1,
   years    = ifelse(df.att$years<=3,0,
              ifelse(df.att$years<=13,1,2)),
   age      = ifelse(df.att$age<=35,0,
                ifelse(df.att$age<=45,1,2)),
   practice = df.att$practice,
   lawschool= df.att$lawschool-1)

# calculate joint entropies
J <- joint_entropy(df.att.ed)
# joint entropy matrix
J$matrix
# frequency distribution of joint entropy values
J$freq
```

lawdata *Law Firm*

Description

This data set comes from a network study of corporate law partnership that was carried out in a Northeastern US corporate law firm, referred to as SG&R, 1988-1991 in New England. It includes (among others) measurements of networks among the 71 attorneys (partners and associates) of this firm, i.e. their strong- co-worker network, advice network, friendship network, and indirect control networks. Various members' attributes are also part of the data set, including seniority, formal status, office in which they work, gender, law school attended. The ethnography, organizational and network analyses of this case are available in Lazega (2001).

Basic advice network: "Think back over the past year, consider all the lawyers in your Firm. To whom did you go for basic professional advice? For instance, you want to make sure that you are handling a case right, making a proper decision, and you want to consult someone whose professional opinions are in general of great value to you. By advice I do not mean simply technical advice."

Friendship network: "Would you go through this list, and check the names of those you socialize with outside work. You know their family, they know yours, for instance. I do not mean all the people you are simply on a friendly level with, or people you happen to meet at Firm functions."

Strong coworkers network: "Because most firms like yours are also organized very informally, it is difficult to get a clear idea of how the members really work together. Think back over the past year, consider all the lawyers in your Firm. Would you go through this list and check the names of those with whom you have worked with. (By "worked with" I mean that you have spent time together on at least one case, that you have been assigned to the same case, that they read or used your work product or that you have read or used their work product; this includes professional work done within the Firm like Bar association work, administration, etc.)"

Usage

```
data(lawdata)
```

Format

List containing the following objects as numbered

1. adjacency matrix for advice seeking (directed)
2. adjacency matrix for friendship (directed)
3. adjacency matrix for cowork (undirected)
4. dataframe with the following attributes on each lawyer:
   - `senior` seniority (ranked from most to least senior)
   - `status` 1=partner; 2=associate
   - `gender` 1=man; 2=woman
   - `office` 1=Boston; 2=Hartford; 3=Providence
   - `years` years with the firm
   - `age` age of attorney
   - `practice` 1=litigation; 2=corporate
   - `lawschool` 1=harvard/yale; 2=ucon; 3= other

   Note: the first 36 out of 71 respondents are the partners in the firm.

Source

https://www.stats.ox.ac.uk/~snijders/siena/Lazega_lawyers_data.htm

References

Emmanuel Lazega, *The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership*, Oxford University Press (2001).

Tom A.B. Snijders, Philippa E. Pattison, Garry L. Robins, and Mark S. Handcock. New specifications for exponential random graph models. *Sociological Methodology* (2006), 99-153.

Examples

```
data(lawdata)
## assign the correct names to the objects in the list
adj.advice <- lawdata[[1]]
adj.friend <- lawdata[[2]]
adj.cowork <-lawdata[[3]]
df.att <- lawdata[[4]]
```

`prediction_power` *Prediction Power*

Description

Computes prediction power when pairs of variables in a given dataframe are used to predict a third variable from the same dataframe. The prediction strength is measured by expected conditional entropies.

Usage

```
prediction_power(var, dat)
```

Arguments

`var`	character string representing the variable in dataframe `dat` to be predicted by pairs of other variables in the dataframe `dat`.
`dat`	dataframe with rows as observations and columns as variables. Variables must all be observed or transformed categorical with finite range spaces.

Details

The expected conditional entropy given by

$$EH(Z|X,Y) = H(X,Y,Z) - H(X, Y)$$

measures the prediction uncertainty when pairs of variables X and Y are used to predict variable Z. The lower the value of EH given different pairs of variables X and Y, the stronger is the prediction of Z.

Value

Upper triangular matrix giving the expected conditional entropies of pairs of variables given as rows and columns of the matrix. The diagonal gives $EH(Z|X) = H(X,Z) - H(X)$, that is when only one variable is used to predict `var`. Note that `NA`'s are in the entire row and column representing the variable being predicted.

Author(s)

Termeh Shafie

References

Frank, O., & Shafie, T. (2016). Multivariate entropy analysis of network data. *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique*, 129(1), 45-63.

Nowicki, K., Shafie, T., & Frank, O. (Forthcoming 2022). *Statistical Entropy Analysis of Network Data*.

See Also

entropy_trivar, entropy_bivar

Examples

```
# use internal data set
data(lawdata)
df.att <- lawdata[[4]]

# three steps of data editing:
# 1. categorize variables 'years' and 'age' based on
# approximately three equally size groups (values based on cdf)
# 2. make sure all outcomes start from the value 0 (optional)
# 3. remove variable 'senior' as it consists of only unique values (thus redundant)
df.att.ed <- data.frame(
   status   = df.att$status,
   gender   = df.att$gender,
   office   = df.att$office-1,
   years    = ifelse(df.att$years<=3,0,
              ifelse(df.att$years<=13,1,2)),
   age      = ifelse(df.att$age<=35,0,
                ifelse(df.att$age<=45,1,2)),
   practice = df.att$practice,
   lawschool= df.att$lawschool-1)

# power of predicting 'status' using pairs of other variables
prediction_power('status', df.att.ed)
```

redundancy *Redundant Variables & Dimensionality Reduction*

Description

Finds redundant variables in a dataframe consisting of discrete variables.

Usage

```
redundancy(dat, dec = 3)
```

Arguments

`dat`	dataframe with rows as observations and columns as variables. Variables must all be observed or transformed categorical with finite range spaces.
`dec`	the precision given as number of decimals used to round bivariate entropies in order to find redundant variables (the more decimals, the harder to detect redundancy). Default is 3.

Details

Redundancy is defined as two variables holding the same information (bivariate entropies) as at least one of the variable alone (univariate entropies). Consider removing one of these two variable from the dataframe for further analysis.

Value

Binary matrix indicating which row and column variables hold the same information.

Author(s)

Termeh Shafie

References

Frank, O., & Shafie, T. (2016). Multivariate entropy analysis of network data. *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique*, 129(1), 45-63.

Nowicki, K., Shafie, T., & Frank, O. (Forthcoming 2022). *Statistical Entropy Analysis of Network Data*.

See Also

`entropy_bivar`,

Examples

```
# use internal data set
data(lawdata)
df.att <- lawdata[[4]]

# two steps of data editing:
# 1. categorize variables 'years' and 'age' based on
# approximately three equally size groups (values based on cdf)
# 2. make sure all outcomes start from the value 0 (optional)
df.att.ed <- data.frame(
   senior   = df.att$senior,
   status   = df.att$status,
   gender   = df.att$gender,
   office   = df.att$office-1,
   years    = ifelse(df.att$years<=3,0,
              ifelse(df.att$years<=13,1,2)),
   age      = ifelse(df.att$age<=35,0,
                ifelse(df.att$age<=45,1,2)),
   practice = df.att$practice,
   lawschool= df.att$lawschool-1)

# find redundant variables in dataframe
redundancy(df.att.ed) # variable 'senior' should be omitted
```

Index