

Package: msu (via r-universe)

August 26, 2024

Title Multivariate Symmetric Uncertainty and Other Measurements

Version 0.0.1

Description Estimators for multivariate symmetrical uncertainty based on the work of Gustavo Sosa et al. (2016) <[arXiv:1709.08730](https://arxiv.org/abs/1709.08730)>, total correlation, information gain and symmetrical uncertainty of categorical variables.

Depends R (>= 3.4.1)

Imports entropy (>= 1.2.1)

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests testthat

Repository <https://eliasmacielr.r-universe.dev>

RemoteUrl <https://github.com/eliasmacielr/msu>

RemoteRef HEAD

RemoteSha 8524d9202c79a328f9bd944c0fa5497cdd08f692

Contents

categorical_sample_size	2
information_gain	2
joint_shannon_entropy	3
msu	4
multivar_joint_shannon_entropy	5
new_informative_variable	6
new_variable	6
new_xor_variables	7
rel_freq	7
sample_size	8
shannon_entropy	9
symmetrical_uncertainty	9
total_correlation	10

Index**12**

`categorical_sample_size`*Estimate the sample size for a variable in function of its categories.*

Description

Estimate the sample size for a variable in function of its categories.

Usage

```
categorical_sample_size(categories, increment = 10)
```

Arguments

<code>categories</code>	A vector containing the number of categories of each variable.
<code>increment</code>	A number as a constant to which the sample size is incremented as a product.

Value

The sample size for a categorical variable based on a ordered permutation heuristic approximation of its categories.

`information_gain`*Estimating information gain between two categorical variables.*

Description

Information gain (also called mutual information) is a measure of the mutual dependence between two variables (see https://en.wikipedia.org/wiki/Mutual_information).

Usage

```
information_gain(x, y)
```

```
IG(x, y)
```

Arguments

<code>x</code>	A factor representing a categorical variable.
<code>y</code>	A factor representing a categorical variable.

Value

Information gain estimation based on Sannon entropy for variables `x` and `y`.

Examples

```
information_gain(factor(c(0,1)), factor(c(1,0)))
information_gain(factor(c(0,0,1,1)), factor(c(0,1,1,1)))
information_gain(factor(c(0,0,1,1)), factor(c(0,1,0,1)))
## Not run:
information_gain(c(0,1), c(1,0))

## End(Not run)
```

joint_shannon_entropy *Estimation of the Joint Shannon entropy for two categorical variables.*

Description

The joint Shannon entropy provides an estimation of the measure of uncertainty between two random variables (see https://en.wikipedia.org/wiki/Joint_entropy).

Usage

```
joint_shannon_entropy(x, y)
```

```
joint_H(x, y)
```

Arguments

x	A factor as the represented categorical variable.
y	A factor as the represented categorical variable.

Value

Joint Shannon entropy estimation for variables x and y.

See Also

[shannon_entropy](#) for the entropy for a single variable and [multivar_joint_shannon_entropy](#) for the entropy associated with more than two random variables.

Examples

```
joint_shannon_entropy(factor(c(0,0,1,1)), factor(c(0,1,0,1)))
joint_shannon_entropy(factor(c('a','b','c')), factor(c('c','b','a')))
## Not run:
joint_shannon_entropy(1)
joint_shannon_entropy(c('a','b'), c('d','e'))

## End(Not run)
```

Description

MSU is a generalization of symmetrical uncertainty (SU) where it is considered the interaction between two or more variables, whereas SU can only consider the interaction between two variables. For instance, consider a table with two variables X1 and X2 and a third variable, Y (the class of the case), that results from the logical XOR operator applied to X1 and X2

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0

For this case

$$MSU(X1, X2, Y) = 0.5.$$

This, in contrast to the measurements obtained by SU of the variables X1 and X2 against Y,

$$SU(X1, Y) = 0$$

and

$$SU(X2, Y) = 0.$$

Usage

```
msu(table_variables, table_class)
```

Arguments

`table_variables`

A list of factors as categorical variables.

`table_class`

A factor representing the class of the case.

Value

Multivariate symmetrical uncertainty estimation for the variable set `{table_variables, table_class}`.

The result is rounded to 7 decimal places.

See Also

[symmetrical_uncertainty](#)

Examples

```
# completely predictable
msu(list(factor(c(0,0,1,1))), factor(c(0,0,1,1)))
# XOR
msu(list(factor(c(0,0,1,1)), factor(c(0,1,0,1))), factor(c(0,1,1,0)))
## Not run:
msu(c(factor(c(0,0,1,1)), factor(c(0,1,0,1))), factor(c(0,1,1,0)))
msu(list(factor(c(0,0,1,1)), factor(c(0,1,0,1))), c(0,1,1,0))

## End(Not run)
```

```
multivar_joint_shannon_entropy
```

Estimation of joint Shannon entropy for a set of categorical variables.

Description

The multivariate joint Shannon entropy provides an estimation of the measure of the uncertainty associated with a set of variables (see https://en.wikipedia.org/wiki/Joint_entropy).

Usage

```
multivar_joint_shannon_entropy(table_variables, table_class)
```

```
multivar_joint_H(table_variables, table_class)
```

Arguments

table_variables

A list of factors as categorical variables.

table_class

A factor representing the class of the case.

Value

Joint Shannon entropy estimation for the variable set table.variables, table.class.

See Also

[shannon_entropy](#) for the entropy for a single variable and [joint_shannon_entropy](#) for the entropy associated with two random variables.

Examples

```
multivar_joint_shannon_entropy(list(factor(c(0,1)), factor(c(1,0))),
  factor(c(1,1)))
```

```
new_informative_variable
```

Create an informative uniform categorical random variable.

Description

The sampling for the items of the created variable is done with replacement.

Usage

```
new_informative_variable(variable_labels, variable_class,
  information_level = 1)
```

Arguments

`variable_labels`

A factor as the labels for the new informative variable.

`variable_class` A factor as the class of the variable.

`information_level`

A integer as the information level of the new variable.

Value

A factor that represents an informative uniform categorical random variable created using the Kononenko method.

```
new_variable
```

Create a uniform categorical random variable.

Description

The sampling for the items of the created variable is done with replacement.

Usage

```
new_variable(elements, n)
```

Arguments

`elements`

A vector with the elements from which to choose to create the variable.

`n`

An integer indicating the number of items to be contained in the variable.

Value

A factor that represents a uniform categorical variable.

Examples

```
new_variable(c(0,1), 4)
new_variable(c('a','b','c'), 10)
```

new_xor_variables *Create a set of categorical variables using the logical XOR operator.*

Description

Create a set of categorical variables using the logical XOR operator.

Usage

```
new_xor_variables(n_variables = 2, n_instances = 1000, noise = 0)
```

Arguments

n_variables	An integer as the number of variables to be created. It is the number of column variables of the table, an additional column is added as a result of the XOR operator over the instances.
n_instances	An integer as the number of instances to be created. It is the number of rows of the table.
noise	A float number as the noise level for the variables.

Value

A set of random variables constructed using the logical XOR operator.

Examples

```
new_xor_variables(2, 4, 0)
new_xor_variables(5, 10, 0.5)
```

rel_freq *Relative frequency of values of a categorical variable.*

Description

Relative frequency of values of a categorical variable.

Usage

```
rel_freq(variable)
```

Arguments

variable A factor as a categorical variable

Value

Relative frequency distribution table for the values in variable.

Examples

```
rel_freq(factor(c(0,1)))
rel_freq(factor(c('a','a','b')))
## Not run:
rel_freq(c(0,1))

## End(Not run)
```

sample_size *Estimate the sample size for a categorical variable.*

Description

Estimate the sample size for a categorical variable.

Usage

```
sample_size(max, min = 1, z = 1.96, error = 0.05)
```

Arguments

max A number as the maximum value of the possible categories.
min A number as the minimum value of the possible categories.
z A number as the confidence coefficient.
error Admissible sampling error.

Value

The sample size for a categorical variable based on a variance heuristic approximation.

shannon_entropy	<i>Estimation of Shannon entropy for a categorical variable.</i>
-----------------	--

Description

The Shannon entropy estimates the average minimum number of bits needed to encode a string of symbols, based on the frequency of the symbols (see http://www.bearcave.com/misl/misl_tech/wavelets/compression/shannon.html).

Usage

```
shannon_entropy(x)
```

$H(x)$

Arguments

x A factor as the represented categorical variable.

Value

Shannon entropy estimation of the categorical variable.

Examples

```
shannon_entropy(factor(c(1,0)))
shannon_entropy(factor(c('a','b','c'))))
## Not run:
shannon_entropy(1)
shannon_entropy(c('a','b','c'))

## End(Not run)
```

symmetrical_uncertainty	<i>Estimating Symmetrical Uncertainty of two categorical variables.</i>
-------------------------	---

Description

Symmetrical uncertainty (SU) is the product of a normalization of the information gain (IG) with respect to entropy. $SU(X,Y)$ is a value in the range $[0,1]$, where $SU(X,Y) = 0$ if X and Y are totally independent and $SU(X,Y) = 1$ if X and Y are totally dependent.

Usage

```
symmetrical_uncertainty(x, y)
```

$SU(x, y)$

Arguments

x A factor as the represented categorical variable.
 y A factor as the represented categorical variable.

Value

Symmetrical uncertainty estimation based on Sannon entropy. The result is rounded to 7 decimal places.

See Also

[msu](#)

Examples

```
# completely predictable
symmetrical_uncertainty(factor(c(0,1,0,1)), factor(c(0,1,0,1)))
# XOR factor variables
symmetrical_uncertainty(factor(c(0,0,1,1)), factor(c(0,1,1,0)))
symmetrical_uncertainty(factor(c(0,1,0,1)), factor(c(0,1,1,0)))
## Not run:
symmetrical_uncertainty(c(0,1,0,1), c(0,1,1,0))

## End(Not run)
```

total_correlation	<i>Estimation of total correlation for a set of categorical random variables.</i>
-------------------	---

Description

Total Correlation is a generalization of information gain (IG) to measure the dependency of a set of categorical random variables (see https://en.wikipedia.org/wiki/Total_correlation).

Usage

```
total_correlation(table_variables, table_class)
```

```
C(table_variables, table_class)
```

Arguments

table_variables A list of factors as categorical variables.
 table_class A factor representing the class of the case.

Value

Total correlation estimation for the variable set `table.variables`, `table.class`.

Examples

```
total_correlation(list(factor(c(0,1)), factor(c(1,0))), factor(c(0,0)))
total_correlation(list(factor(c('a','b')), factor(c('a','b'))),
  factor(c('a','b')))
## Not run:
total_correlation(list(factor(c(0,1)), factor(c(1,0))), c(0,0))
total_correlation(c(factor(c(0,1)), factor(c(1,0))), c(0,0))

## End(Not run)
```

Index

C (total_correlation), 10
categorical_sample_size, 2

H (shannon_entropy), 9

IG, 9, 10
IG (information_gain), 2
information_gain, 2

joint_H (joint_shannon_entropy), 3
joint_shannon_entropy, 3, 5

msu, 4, 10
multivar_joint_H
 (multivar_joint_shannon_entropy),
 5
multivar_joint_shannon_entropy, 3, 5

new_informative_variable, 6
new_variable, 6
new_xor_variables, 7

rel_freq, 7

sample_size, 8
shannon_entropy, 3, 5, 9
SU, 4
SU (symmetrical_uncertainty), 9
symmetrical_uncertainty, 4, 9

total_correlation, 10