Package 'gorica'

December 13, 2019

Type Package

Date 2019-11-24

Title Evaluation of Inequality Constrained Hypotheses Using GORICA

Version 0.1.0

Description Implements the generalized order-restricted information criterion approximation (GORICA) by Kuiper, Hoijtink, & Silvapulle (2012) <doi:10.1016/j.jspi.2012.03.007>. The GORICA is an AIC-like information criterion that
can be utilized to evaluate informative hypotheses specifying directional relationships between model parameters in terms of (in)equality constraints. The GORICA is applicable not only to
normal linear models, but also to generalized linear models (GLMs), generalized linear mixed models (GLMMs), and structural equation models (SEMs). In addition, the GORICA can be utilized in the context of contingency tables for which (in)equality constrained hypotheses do not necessarily contain linear restrictions on cell probabilities, but instead often contain non-linear restrictions on cell probabilities.

License GPL (>= 3)

Encoding UTF-8

LazyData true

URL https://informative-hypotheses.sites.uu.nl/software/goric/

NeedsCompilation no

RoxygenNote 6.1.1

Depends R (>= 3.0.0), bain (>= 0.2.2), stats

Imports MASS, mvtnorm, quadprog, lme4, lavaan

Suggests testthat

Author Rebecca M. Kuiper [aut],
Altinisik Yasin [aut],
Caspar J. van Lissa [aut, cre]

Maintainer Caspar J. van Lissa <c.j.vanlissa@uu.nl>

Repository CRAN

Date/Publication 2019-12-13 14:20:09 UTC

R topics documented:

`academic_awards` *Academic awards data*

Description

Simulated dataset from https://stats.idre.ucla.edu/sas/dae/poisson-regression/.

Usage

```
data(academic_awards)
```

Format

A data frame with 200 rows and 4 variables.

Details

num_awards	`integer`	Outcome variable; indicates the number of awards earned by students at a high school in a year
math	`integer`	Continuous predictor variable; represents students' scores on their math final exam
prog	`factor`	Categorical predictor variable with three levels, indicating the type of program in which the studen

`gorica` *Evaluate informative hypotheses using the GORICA*

Description

GORICA is an acronym for "generalized order-restricted information criterion approximation". It can be utilized to evaluate informative hypotheses, which specify directional relationships between model parameters in terms of (in)equality constraints.

Usage

```
gorica(x, hypothesis, comparison = "unconstrained", ...)
```

Arguments

x An R object containing the outcome of a statistical analysis. Currently, the following objects can be processed:

- lm() objects (anova, ancova, multiple regression).
- t_test() objects.
- lavaan objects.
- lmerMod objects.
- A named vector containing the estimates resulting from a statistical analysis, when the argument Sigma is also specified. Note that, named means that each estimate has to be labeled such that it can be referred to in hypotheses.

hypothesis A character string containing the informative hypotheses to evaluate (see Details).

comparison A character string indicating what the hypothesis should be compared to. Defaults to comparison = "unconstrained"; options include c("unconstrained","complement","none'

... Additional arguments passed to the internal function compare_hypotheses.

Details

The GORICA is applicable to not only normal linear models, but also applicable to generalized linear models (GLMs) (McCullagh & Nelder, 1989), generalized linear mixed models (GLMMs) (McCullogh & Searle, 2001), and structural equation models (SEMs) (Bollen, 1989). In addition, the GORICA can be utilized in the context of contingency tables for which (in)equality constrained hypotheses do not necessarily contain linear restrictions on cell probabilities, but instead often contain non-linear restrictions on cell probabilities.

hypotheses is a character string that specifies which informative hypotheses have to be evaluated. A simple example is hypotheses <-"a > b > c; a = b = c;" which specifies two hypotheses using three estimates with names "a", "b", and "c", respectively.

The hypotheses specified have to adhere to the following rules:

1. Parameters are referred to using the names specified in names().
2. Linear combinations of parameters must be specified adhering to the following rules:
   (a) Each parameter name is used at most once.
   (b) Each parameter name may or may not be pre-multiplied with a number.
   (c) A constant may be added or subtracted from each parameter name.
   (d) A linear combination can also be a single number.

   Examples are: 3 * a + 5; a + 2 * b + 3 * c -2; a -b; and 5.
3. (Linear combinations of) parameters can be constrained using <, >, and =. For example, a > 0 or a > b = 0 or 2 * a < b + c > 5.
4. The ampersand & can be used to combine different parts of a hypothesis. For example, a > b & b > c which is equivalent to a > b > c or a > 0 & b > 0 & c > 0.
5. Sets of (linear combinations of) parameters subjected to the same constraints can be specified using (). For example, a > (b,c) which is equivalent to a > b & a > c.
6. The specification of a hypothesis is completed by typing ; For example, hypotheses <-"a > b > c; a = b = c;", specifies two hypotheses.

7. Hypotheses have to be compatible, non-redundant and possible. What these terms mean will be elaborated below.

The set of hypotheses has to be compatible. For the statistical background of this requirement see Gu, Mulder, Hoijtink (2018). Usually the sets of hypotheses specified by researchers are compatible, and if not, `gorica` will return an error message. The following steps can be used to determine if a set of hypotheses is compatible:

1. Replace a range constraint, e.g., `1 < a1 < 3`, by an equality constraint in which the parameter involved is equated to the midpoint of the range, that is, `a1 = 2`.
2. Replace in each hypothesis the < and > by =. For example, `a1 = a2 > a3 > a4` becomes `a1 = a2 = a3 = a4`.
3. The hypotheses are compatible if there is at least one solution to the resulting set of equations. For the two hypotheses considered under 1. and 2., the solution is a1 = a2 = a3 = a4 = 2. An example of two non-compatible hypotheses is `hypotheses <-"a = 0; a > 2;"` because there is no solution to the equations `a=0` and `a=2`.

Each hypothesis in a set of hypotheses has to be non-redundant. A hypothesis is redundant if it can also be specified with fewer constraints. For example, `a = b & a > 0 & b > 0` is redundant because it can also be specified as `a = b & a > 0`. `gorica` will work correctly if hypotheses specified using only < and > are redundant. `gorica` will return an error message if hypotheses specified using at least one = are redundant.

Each hypothesis in a set of hypotheses has to be possible. An hypothesis is impossible if estimates in agreement with the hypothesis do not exist. For example: values for `a` in agreement with `a = 0 & a > 2` do not exist. It is the responsibility of the user to ensure that the hypotheses specified are possible. If not, `gorica` will either return an error message or render an output table containing `Inf`'s.

Value

An object of class `gorica`, containing the following elements:

- `fit` A `data.frame` containing the loglikelihood, penalty (for complexity), the GORICA value, and the GORICA weights. The GORICA weights are calculated by taking into account the misfits and complexities of the hypotheses under evaluation. These weights are used to quantify the support in the data for each hypothesis under evaluation. By looking at the pairwise ratios between the GORICA weights, one can determine the relative importance of one hypothesis over another hypothesis.
- `call` The original function call.
- `model` The original model object (`x`).
- `estimates` The parameters extracted from the `model`.
- `Sigma` The asymptotic covariance matrix of the `estimates`.
- `comparison` Which alternative hypothesis was used.
- `hypotheses` The hypotheses evaluated in `fit`.

Author(s)

Caspar van Lissa, Yasin Altinisik, Rebecca Kuiper

References

Altinisik, Y. (2018). Evaluation of Inequality Constrained Hypotheses Using an Akaike-Type Information Criterion (Doctoral dissertation, Utrecht University). ISBN: 978-90-393-6918-0. https://dspace.library.uu.nl/handle/1874/360604

Bollen, K. (1989). Structural equations with latent variables. New York, NY: John Wiley and Sons.

Kuiper, R. M., Hoijtink, H., & Silvapulle, M. J. (2011). An Akaike-type information criterion for model selection under inequality constraints. Biometrika, 98, 495-501. doi:10.1093/biomet/asr002

Kuiper, R. M., Hoijtink, H., & Silvapulle, M. J. (2012). Generalization of the order-restricted information criterion for multivariate normal linear models. Journal of statistical planning and inference, 142(8), 2454-2463. doi:10.1016/j.jspi.2012.03.007

McCullagh, P. & Nelder, J. (1989). Generalized linear models (2nd ed.). Boca Raton, FL: Chapman & Hall / CRC.

McCullogh, C. E., & Searle, S. R. (2001). Generalized linear and mixed models. New York, NY: Wiley.

Examples

```
# EXAMPLE 1. One-sample t test
ttest1 <- t_test(iris$Sepal.Length,mu=5)
gorica(ttest1,"x<5.8")

# EXAMPLE 2. ANOVA
aov1 <- aov(yield ~ block-1 + N * P + K, npk)
gorica(aov1,hypothesis="block1=block5;
   K1<0")

# EXAMPLE 3. gml
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)
fit <- glm(counts ~ outcome-1 + treatment, family = poisson())
gorica(fit, "outcome1 > (outcome2, outcome3)")

# EXAMPLE 4. ANOVA
res <- lm(Sepal.Length ~ Species-1, iris)
est <- get_estimates(res)
est
gor <- gorica(res, "Speciessetosa < (Speciesversicolor, Speciesvirginica)",
comparison = "complement")
gor
```

`reading_ach` *Reading achievement data*

Description

Dataset based on Finch, Bolin, and Kelley (2014, p.32).

Usage

```
data(reading_ach)
```

Format

A data frame with 10320 rows and 5 variables.

Details

school	`integer`	Clustering variable representing the school a given participant was enrolled in
gender	`factor`	Binary factor variable representing participants' assigned sex
age	`integer`	Participants' age in months
geread	`numeric`	Reading achievement
gevocab	`numeric`	Vocabulary

References

Finch, W. H., Bolin, J. E., & Kelley, K. (2014). Multilevel modeling using r. CRC Press 2014.

`wechsler` *Wechsler intelligence test data*

Description

Dataset based on McArdle and Prescott (1992, p.90). This study evaluates intelligence and cognitive ability in a sample of individuals over 18 years of age (N = 1680) using the IQ test Wechsler Adult Intelligence Scale-Revised (WAIS-R) (Wechsler, 1981).

Usage

```
data(wechsler)
```

Format

A data frame with 1680 rows and 10 variables.

Details

age	`integer`	Participants' age (recoded)
edc	`factor`	Whether a participant graduated high school or not (1 = not graduated, 2 = graduated)
y1	`integer`	information; general knowledge of participants
y2	`integer`	comprehension; ability of abstract reasoning or judgment
y3	`integer`	similarities; unifying a theme
y4	`integer`	vocabulary; verbal definition
y5	`integer`	picture completion; perceiving visual images with missing features
y6	`integer`	block design; arranging blocks to match a design
y7	`integer`	picture arrangement; ordering cards with true story lines
y8	`integer`	object assembly; reassembling puzzles

References

McArdle, J. J., & Prescott, C. A. (1992). Age-based construct validation using structural equation modeling. Experimental Aging Research, 18, 87-115.

Index