Package: cotram (via r-universe)

Title Count Transformation Models

Version 0.5-2

Date 2024-09-18

Description Count transformation models featuring parameters
interpretable as discrete hazard ratios, odds ratios,
reverse-time discrete hazard ratios, or transformed
expectations. An appropriate data transformation for a count
outcome and regression coefficients are simultaneously
estimated by maximising the exact discrete log-likelihood using
the computational framework provided in package 'mlt',
technical details are given in Siegfried & Hothorn (2020)
<DOI:10.1111/2041-210X.13383>. The package also contains an
experimental implementation of multivariate count
transformation models with an application to multi-species
distribution models <DOI:10.48550/arXiv.2201.13095>.

Depends R (>= 3.5.0), tram (>= 1.0-0), mlt (>= 1.4-7)

Imports variables (>= 1.0-2), basefun (>= 1.0-5), alabama, Matrix,
methods, survival, utils, qrng

Suggests TH.data, knitr, lattice, colorspace, multcomp, mvtnorm,
numDeriv

VignetteBuilder knitr

URL http://ctm.R-forge.R-project.org

Encoding UTF-8

License GPL-2

Repository https://trafo-universe.r-universe.dev

RemoteUrl https://github.com/r-forge/ctm

RemoteRef HEAD

RemoteSha bf68b8828a608dd9499a74af1bfdb746420dbc03

Contents

---

confband | *Confidence Bands*

Description

Confidence bands for transformation, distribution, survivor or cumulative hazard functions

Usage

```
## S3 method for class 'cotram'
confband(object, newdata,  level = 0.95,
       type = c("trafo", "distribution", "survivor", "cumhazard"),
       smooth = FALSE, q = NULL, K = 20, cheat = K, ...)
```

Arguments

object	an object of class [cotram]().
newdata	a data frame of observations.
level	the confidence level.
type	the function to compute the confidence band for.
smooth	logical; if TRUE a smoothed function of type is returned.
q	quantiles at which to evaluate the model.
K	number of grid points the function is evaluated at (in the absence of q and smooth = TRUE).
cheat	number of grid points the function is evaluated at when using the quantile obtained for K grid points (in the absence of q and smooth = TRUE).
...	additional arguments to [confint.glht]().

Details

The function is evaluated at the count response or at K grid points and simultaneous confidence intervals are then interpolated in order to construct the band.

Value

For each row in newdata the function and corresponding confidence band evaluated at the count response (or K or cheat grid points) is returned.

Examples

```
op <- options(digits = 4)

data("birds", package = "TH.data")

### fit count transformation model with cloglog link
m_birds <- cotram(SG5 ~ AOT + AFS + GST + DBH + DWC + LOG, data = birds,
                  method = "cloglog")

### compute asymptotic confidence bands for the distribution function
### for the first oberservation
confband(m_birds, newdata = birds[1, ], type = "distribution")

options(op)
```

---

cotram   *Count Transformation Models*

Description

Likelihood-based count transformation models for fully parameterised discrete conditional distribution functions. The link function governing the interpretation of the predictor can be chosen and results in discrete hazard ratios, odds ratios, reverse time hazard ratios or conditional expectation of transformed counts.

Usage

```
cotram(formula, data, method = c("logit", "cloglog", "loglog", "probit"),
       log_first = TRUE, prob = 0.9, subset, weights, offset, cluster,
       na.action = na.omit, ...)
```

Arguments

formula   an object of class "formula": a symbolic description of the model structure to
              be fitted. The details of model specification are given under tram and in the
              package vignette.

data   an optional data frame, list or environment (or object coercible by as.data.frame
              to a data frame) containing the variables in the model. If not found in data, the
              variables are taken from environment(formula).

method   character specifying the choice of the link function, mapping the transformation
              function into probabilities. Available choices include the logit, complementary
              log-log, log-log or probit link. The different link functions govern the interpre-
              tation of the linear predictor. Details of the interpretation can be found in the
              package vignette.

prob	probability giving the quantile of the response defining the upper limit of the support of a smooth Bernstein polynomial (with the lower limit being set to 0). If a vector of two probabilites is specified, the corresponding quantiles of the response define the lower and upper limit of the support, respectively. Note, that the support is rounded to integer values.
log_first	logical; if TRUE, a Bernstein polynomial is defined on the log-scale for (y + 1).
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. If present, the weighted log-likelihood is maximised.
offset	this can be used to specify an _a priori_ known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases.
cluster	optional factor with a cluster ID employed for computing clustered covariances.
na.action	a function which indicates what should happen when the data contain NAs. The default is set to na.omit.
...	additional arguments to [tram]().

Details

Likelihood-based estimation of a fully parameterised conditional discrete distribution function for count data, while ensuring interpretability of the linear predictors. The models are defined with a negative shift term relating positive predictors to larger values of the conditional mean. For the model with logistic or cloglog link exp(-coef()) is the multiplicative change of discrete odds-ratios or hazard ratios. For the model with loglog link exp(coef()) is the multiplicative change of the reverse time hazard ratios. Applying a transformation model with probit link coef() gives the conditional expectation of the transformed counts, with transformation function estimated from data.

Value

An object of class cotram and tram, with corresponding coef, vcov, logLik, summary, print, plot and predict methods.

References

Sandra Siegfried, Torsten Hothorn (2020), Count Transformation Models, *Methods in Ecology and Evolution*, **11**(7), 818–827, doi:10.1111/2041210X.13383.

Torsten Hothorn, Lisa Möst, Peter Bühlmann (2018), Most Likely Transformations, *Scandinavian Journal of Statistics*, **45**(1), 110–134, doi:10.1111/sjos.12291.

Torsten Hothorn (2020), Most Likely Transformations: The mlt Package, *Journal of Statistical Software*, **92**(1), 1–68, doi:10.18637/jss.v092.i01.

Examples

```
op <- options(digits = 2)

data("birds", package = "TH.data")
cotram(SG5 ~ AOT + AFS + GST + DBH + DWC + LOG, data = birds)

options(op)
```

---

cotram-methods	*Methods for Count Transformation Models*

Description

Methods for objects inheriting from class cotram

Usage

```
## S3 method for class 'cotram'
predict(object, newdata = model.frame(object),
        type = c("lp", "trafo", "distribution", "survivor", "density",
                   "logdensity", "hazard", "loghazard", "cumhazard",
                   "logcumhazard", "odds", "logodds", "quantile"),
        smooth = FALSE, q = NULL, K = 20, prob = 1:(10-1)/10, ...)
## S3 method for class 'cotram'
plot(x, newdata, type = c("distribution", "survivor","density",
                            "logdensity", "cumhazard", "quantile", "trafo"),
        confidence = c("none", "band"), level = 0.95,
        smooth = FALSE, q = NULL, K = 20, cheat = K, prob = 1:(10-1)/10,
        col = "black", fill = "lightgrey",
        lty = 1, lwd = 1, add = FALSE, ...)
## S3 method for class 'cotram'
as.mlt(object)
## S3 method for class 'cotram'
logLik(object, parm = coef(as.mlt(object), fixed = FALSE), newdata, ...)
```

Arguments

object, x	a fitted linear count transformation model inheriting from class cotram.
newdata	an optional data frame of observations.
parm	model parameters.
type	type of prediction, current options include linear predictors ("lp", of x variables in the formula y ~ x), transformation functions ("trafo") or distribution functions on the scale of the cdf ("distribution"), survivor function, density function, log-density function, cumulative hazard function or quantile function.
confidence	whether to plot a confidence band (see [confband](#)).

level	the confidence level.
smooth	logical; if TRUE a smoothed function of type is returned.
q	quantiles at which to evaluate the model.
prob	probabilities for the evaluation of the quantile function (type = "quantile").
K	number of grid points the function is evaluated at (for smooth = TRUE and in absence of q).
cheat	number of grid points the function is evaluated at when using the quantile obtained for K grid points (in the absence of q and smooth = TRUE).
col	color for the lines to plot.
fill	color for the confidence band.
lty	line type for the lines to plot.
lwd	line width.
add	logical; indicating if a new plot shall be generated (the default).
...	additional arguments to the underlying methods for predict.

Details

predict and plot can be used to inspect the model on different scales.

See Also

predict.cotram, confband.cotram, tram-methods, mlt-methods, plot.ctm

Examples

```
op <- options(digits = 4)

data("birds", package = "TH.data")

### fit count transformation model with cloglog link
m_birds <- cotram(SG5 ~ AOT + AFS + GST + DBH + DWC + LOG, data = birds,
                  method = "cloglog")
logLik(m_birds)

### classical likelihood inference
summary(m_birds)

### coefficients of the linear predictor (discrete hazard ratios)
exp(-coef(m_birds))

### compute predicted median along with 10% and 90% quantile for the first
### three observations
nd <- birds[1:3,]
round(predict(m_birds, newdata = nd, type = "quantile", prob = c(.1, .5, .9),
              smooth = TRUE), 3)

### plot the predicted distribution for these observations
plot(m_birds, newdata = nd, type = "distribution",
```

```
        col = c("skyblue", "grey", "seagreen"))

    options(op)
```

---

mcotram   *Multivariate Count Conditional Transformation Models*

Description

A proof-of-concept implementation of multivariate conditional transformation models for count data.

Usage

```
mcotram(..., formula = ~ 1, data, conditional = FALSE, theta = NULL,
        fixed = NULL, scale = FALSE, optim = mmltoptim(),
        M = 1000, dofit = TRUE, domargins = TRUE)
```

Arguments

...	marginal count transformation models, one for each response
formula	a model formula describing a model for the dependency structure via the lambda parameters. The default is set to ~ 1 for constant lambdas.
data	a data.frame.
conditional	logical; parameters are defined conditionally (only possible when all models are probit models). This is the default as described by Klein et al. (2022). If FALSE, parameters can be directly interpreted marginally, this is explained in Section 2.6 by Klein et al. (2022). Using conditional = FALSE with probit-only models gives the same likelihood but different parameter estimates.
theta	an optional vector of starting values.
fixed	an optional named numeric vector of predefined parameter values.
scale	a logical indicating if (internal) scaling shall be applied to the model coefficients.
optim	a list of optimisers as returned by [mmltoptim](#)
M	number of Halton sequences used to approximate the log-likelihood in [lpmvnorm](#).
dofit	logical; parameters are fitted by default, otherwise a list with log-likelihood and score function is returned.
domargins	logical; all model parameters are fitted by default, including the parameters of marginal models.

Details

The function implements multivariate count conditional transformation models. The response is assumed to be a vector of counts.

Value

An object of class `mmlt` with `coef` and `predict` methods.

References

Luisa Barbani, Roland Brandl, Torsten Hothorn (2022), Multi-species Count Transformation Models, doi:10.48550/arXiv.2201.13095.

Nadja Klein, Torsten Hothorn, Luisa Barbanti, Thomas Kneib (2020), Multivariate Conditional Transformation Models. *Scandinavian Journal of Statistics*, doi:10.1111/sjos.12501.

Examples

```
library("cotram")
data("spiders", package = "cotram")

### for illustration only
OR <- 1      ### order of transformation function
             ### OR = 1 means log-linear, use OR ~ 6
M <- 100     ### number of Halton sequences, seem sufficient here

## fit conditional marginal count transformation models
## one for each species
## (don't test as it takes too long for CRAN)
m_PF <- cotram(Pardosa_ferruginea ~ Elevation + Canopy_openess,
               data = spiders, method = "probit", order = OR)
m_HL <- cotram(Harpactea_lepida ~ Elevation + Canopy_openess,
               data = spiders, method = "probit", order = OR)
m_CC <- cotram(Callobius_claustrarius ~ Elevation + Canopy_openess,
               data = spiders, method = "probit", order = OR)
m_CT <- cotram(Coelotes_terrestris ~ Elevation + Canopy_openess,
               data = spiders, method = "probit", order = OR)
m_PL <- cotram(Pardosa_lugubris ~ Elevation + Canopy_openess,
               data = spiders, method = "probit", order = OR)
m_PR <- cotram(Pardosa_riparia ~ Elevation + Canopy_openess,
               data = spiders, method = "probit", order = OR)

### fit dependence parameters
mm <- mcotram(m_PF, m_HL, m_CC, m_CT, m_PL, m_PR, data = spiders,
              M = M, scale = TRUE)
logLik(mm)

### Kendall's tau: Dependence of species after accounting
### for elevation and canopy openess in marginal models
coef(mm, type = "Kendall")

### regress dependencies on elevation and canopy openess
mmc <- mcotram(m_PF, m_HL, m_CC, m_CT, m_PL, m_PR, data = spiders,
               formula = ~ Elevation + Canopy_openess, M = M, scale = TRUE)
logLik(mmc)

### weak evidence for such effects
```

```
pchisq(2 * (logLik(mmc) - logLik(mm)), df = 30, lower.tail = FALSE)

### plot Kendall's tau for different elevations / openess levels
nd <- expand.grid(Elevation = 80:120 * 10, Canopy_openess = 1:10 * 10)
KD <- Lower_tri(coef(mmc, newdata = nd, type = "Kendall"))
f <- factor(rownames(KD))
nd <- cbind(f = rep(f, nrow(nd)), nd[rep(1:nrow(nd), each = nlevels(f)),])
nd$KD <- c(KD)

if (require("lattice")) {
contourplot(KD ~ Elevation + Canopy_openess | f, data = nd,
            cuts = 18, xlab = "Elevation", ylab = "Canopy openess")
}


### for example:
### => constant negative dependence of Pardosa_lugubris and Coelotes_terrestris
### => weak dependence of Harpactea_lepida and Pardosa_ferruginea
###    for low elevations, negative dependence increasing with elevation
```

---

spiders   *Bavarian Forest Spider Data*

Description

Abundance of six spider species on 190 plots in the Bavarian Forest National Park

Usage

```
data("spiders")
```

Format

A data frame with 190 observations on the following 9 variables.

Plot experimental plot id

Elevation elevation of the plot

Canopy_openess openess of canopy at plot

Pardosa_ferruginea number of Pardosa ferruginea observed

Harpactea_lepida number of Harpactea lepida observed

Callobius_claustrarius number of Callobius claustrarius observed

Coelotes_terrestris number of Coelotes terrestris observed

Pardosa_lugubris number of Pardosa lugubris observed

Pardosa_riparia number of Pardosa riparia observed

Details

To untangle effects of dead-wood addition and canopy openness on non-saproxylic epigeal arthropods, Seibold et al (2016) exposed different amounts of logs and branches on 190 0.1-ha plots located in sunny or shady mixed montane forests and sampled epigeal arthropods over three years. The data contain abundances of six spider species for all 190 plots, along with plot elevation and canopy openess.

Source

Sebastian Seibold, Claus Bässler, Petr Baldrian, Lena Reinhard, Simon Thorn, Michael D. Ulyshen, Ingmar Weiß and Jörg Müller (2016). Dead-wood addition promotes non-saproxylic epigeal arthropods but effects are mediated by canopy openness. *Biological Conservation*, **204**, 181–188, doi:10.1016/j.biocon.2016.09.031.

Index