Package: multipleDL (via r-universe)

October 29, 2026

Version 1.0.0

Date 2023-11-17

Title Addressing Detection Limits by Cumulative Probability Models (CPMs)

Description Build CPMs (cumulative probability models, also known as cumulative link models) to account for detection limits (both single and multiple detection limits) in response variables. Conditional quantiles and conditional CDFs can be calculated based on fitted models. The package implements methods described in Tian, Y., Li, C., Tu, S., James, N. T., Harrell, F. E., & Shepherd, B. E. (2022). ``Addressing Detection Limits with Semiparametric Cumulative Probability Models". <doi:10.48550/arXiv.2207.02815>.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.1.2

Biarch true

Depends R (>= 3.4.0)

Imports methods, stats, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), rstantools (>= 2.1.1), SparseM

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

NeedsCompilation yes

Author Yuqi Tian [aut, cre], Chun Li [aut], Shengxin Tu [aut], Nathan James [aut], Frank Harrell [aut], Bryan Shepherd [aut]

Maintainer Yuqi Tian <yuqitian35@gmail.com>

Config/pak/sysreqs make

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

Date/Publication 2023-11-25 02:33:25 UTC

RemoteUrl https://github.com/cran/multipleDL

RemoteRef HEAD

RemoteSha f59d7f50aa62e074fee4faa45cc87b28c8e300f4

Contents

multipleDL-package 2
cdf_dl 3
func_link 4
func_link_num 4
func_V 5
multipleDL 5
quantile_dl 7
selectedQr 8

Index **9**

`multipleDL-package` *Address Detection Limits by Cumulative Probability Models*

Description

The package allows fitting regression models on continuous/ordinal response data subject to detection limits (DLs) based on cumulative probability models (CPMs). Both single and multiple DLs can be handled. Conditional quantiles and CDFs (cumulative distribution functions) can obtained from fitted models.

Details

The 'multipleDL' package.

References

Stan Development Team (2020). RSroxygen2::roxygenize()tan: the R interface to Stan. R package version 2.19.3. https://mc-stan.org Harrell, F. (2020). rms: Regression modeling strategies. R package version 6.1.0. https://CRAN.R-project.org/package=rms Tian et al. "Addressing detection limits by semiparametric cumulative probability models." (2022) (to be submitted)

cdf_dl *Calculate conditional CDFs*

Description

This functions calculates the conditional CDFs based on the fitted model and new data.

Usage

```
cdf_dl(mod, new.data, at.y = 0, se = TRUE)
```

Arguments

mod	the model
new.data	the new data
at.y	a numeric vector of cut-off points P(y <= at.y | new.data)
se	if confidence intervals needed (default = TRUE)

Value

A list containing the following components:

est	a vector of estimated condtional CDFs
se	a vector of estimated standard errors
lb	a vector of estimated lower bounds of 95% confidence intervals
ub	a vector of estimated upper bounds of 95% confidence intervals

Examples

```
#' @examples
## Multiple DLs
## generate a small example data: 3 sites with different lower and upper DLs
## lower DLs: site 1: - 0.2; site 2: 0.3; site 3: no lower DL
## upper DLs: site 1: no upper DL; site 2: 4; site 3: 3.5
## each site includes 100 subjects
n <- 100
x <- rnorm(n * 3)
e <- rnorm(n * 3)
y <- exp(x + e)
no_dl <- 1e6
data <- data.frame(y = y, x = x, subset = rep(c(1, 2, 3), each=n))
data$dl_l <- ifelse(data$subset == 1, 0.2, ifelse(data$subset == 2, 0.3, -no_dl))
data$dl_u <- ifelse(data$subset == 1, no_dl, ifelse(data$subset == 2, 4, 3.5))
data$delta_l <- ifelse(data$y >= data$dl_l, 1, 0)
data$delta_u <- ifelse(data$y <= data$dl_u, 1, 0)
data$z <- ifelse(data$delta_l == 0, data$dl_l, ifelse(data$delta_u == 0, data$dl_u, data$y))
# model
```

```
mod <- multipleDL(formula = z ~ x, data = data,
                  delta_lower = data$delta_l, delta_upper = data$delta_u, link='probit')
# new data
new.data <- data.frame(x = c(0, 1))
conditional_median <- quantile_dl(mod, new.data, probs = 0.5)
conditional_cdf <- cdf_dl(mod, new.data, at.y = 1.5) # P(y <= 1.5 | new.data)
```

`func_link` *Link functions*

Description

This function includes necessary functions related to each link function

Usage

```
func_link(link)
```

Arguments

`link`	the link function

Value

A list of functions subject to a link function

`func_link_num` *Link functions (number)*

Description

This function faciliates the stan code (used as an internal function)

Usage

```
func_link_num(link)
```

Arguments

`link`	the link function

Value

An integer representing corresponding link function

func_V *Calculate the covariance matrix*

Description

This functions calculates the covariance matrix based on the point estimates

Usage

```
func_V(coef, n, x, y, delta, k, p, fam)
```

Arguments

coef	coefficients (alpha, beta)
n	number of subjects
x	original covariate matrix
y	ranks of code values
delta	censoring indicators
k	the number of unique code values
p	the number of covariates
fam	a list of functions subject to the link function

Value

A covariance matrix of coefficients

multipleDL *CPMs for multiple detection limits*

Description

This function build the CPM for multiple detection limits (DLs).

Usage

```
multipleDL(formula, data, delta_lower = NULL, delta_upper = NULL, link)
```

Arguments

formula	an R formula object
data	a data frame including response data and covariates
delta_lower	(optional) indicators of lower DLs censoring (1: observed; 0:censored). If not specified, treat as observed.
delta_upper	(optional) indicators of upper DLs censoring(1: observed; 0:censored). If not specified, treat as observed.
link	the link function (probit, logit, loglog, cloglog)

Details

When there are multiple DLs, we appropriately modify the CPM likelihood. If a value is below a lower DL, set the censored value as the lower DL and set the lower DL indicator `delta_lower` to be 0. Similarly, if a value is above an upper DL, set the censored value as the upper DL and set the upper DL indicator `delta_upper` to be 0. This function also works when there is only a single lower and/or upper DL.

Conditional quantiles and CDFs and corresponding 95% confidence intervals can be calculated from the model fit.

Value

A list containing the following components:

`coef`	a numeric vector of estimated coeffiencts
`var`	covariance matrix of estimated coeffiencts
`yunique`	a numeric vector of unique response values
`kint`	number of alphas (intercept terms)
`p`	number of betas (regression coeffiencts)
`fam`	a list of functions associated with the specified link function
`x`	the design matrix
`log_likelihood`	the log-likelihood

References

Tian, Y., Li, C., Tu, S., James, N. T., Harrell, F. E., & Shepherd, B. E. (2022). Addressing Detection Limits with Semiparametric Cumulative Probability Models. arXiv preprint arXiv:2207.02815.

Stan Development Team (2020). RSroxygen2::roxygenize()tan: the R interface to Stan. R package version 2.19.3. https://mc-stan.org

Harrell, F. (2020). rms: Regression modeling strategies. R package version 6.1.0. https://CRAN.R-project.org/package=rms

See Also

`cdf_dl`, `quantile_dl`

Examples

```
## Multiple DLs
## generate a small example data: 3 sites with different lower and upper DLs
## lower DLs: site 1: - 0.2; site 2: 0.3; site 3: no lower DL
## upper DLs: site 1: no upper DL; site 2: 4; site 3: 3.5
## each site includes 100 subjects
n <- 100
x <- rnorm(n * 3)
e <- rnorm(n * 3)
y <- exp(x + e)
no_dl <- 1e6
```

```
data <- data.frame(y = y, x = x, subset = rep(c(1, 2, 3), each=n))
data$dl_l <- ifelse(data$subset == 1, 0.2, ifelse(data$subset == 2, 0.3, -no_dl))
data$dl_u <- ifelse(data$subset == 1, no_dl, ifelse(data$subset == 2, 4, 3.5))
data$delta_l <- ifelse(data$y >= data$dl_l, 1, 0)
data$delta_u <- ifelse(data$y <= data$dl_u, 1, 0)
data$z <- ifelse(data$delta_l == 0, data$dl_l, ifelse(data$delta_u == 0, data$dl_u, data$y))
# model
mod <- multipleDL(formula = z ~ x, data = data,
                  delta_lower = data$delta_l, delta_upper = data$delta_u, link='probit')
# new data
new.data <- data.frame(x = c(0, 1))
conditional_median <- quantile_dl(mod, new.data, probs = 0.5)
conditional_cdf <- cdf_dl(mod, new.data, at.y = 1.5) # P(y <= 1.5 | new.data)


## Single DL: lower DL at 0.5
n <- 100
x <- rnorm(n)
e <- rnorm(n)
y <- exp(x + e)
lower_dl <- 0.5
data <- data.frame(y = y, x = x)
data$delta_lower <- ifelse(data$y >= lower_dl, 1, 0)
data$z <- ifelse(data$delta_lower == 0, lower_dl, data$y)
mod <- multipleDL(formula = z ~ x, data = data,
                   delta_lower = data$delta_l, link='probit')
```

`quantile_dl` *Calculate conditional quantiles*

Description

This functions calculates the conditional weighted quantiles based on the fitted model and new data.

Usage

```
quantile_dl(mod, new.data, probs = 0.5, se = TRUE)
```

Arguments

`mod`	the model
`new.data`	the new data
`probs`	a numeric vector of pth quantiles
`se`	if confidence intervals needed (default = TRUE)

Value

A list containing the following components:

est	a vector of estimated condtional quantiles
lb	a vector of estimated lower bounds of 95% confidence intervals
ub	a vector of estimated upper bounds of 95% confidence intervals

---

selectedQr *QR Decomposition Preserving Selected Columns*

Description

Runs a matrix through the QR decomposition and returns the transformed matrix and the forward and inverse transforming matrices R, Rinv. If columns of the input matrix X are centered the QR transformed matrix will be orthogonal. This is helpful in understanding the transformation and in scaling prior distributions on the transformed scale. not can be specified to keep selected columns as-is. cornerQr leaves the last column of X alone (possibly after centering). When not is specified, the square transforming matrices have appropriate identity submatrices inserted so that recreation of original X is automatic.

Usage

```
selectedQr(X, not = NULL, corner = FALSE, center = TRUE)
```

Arguments

X	a numeric matrix
not	an integer vector specifying which columns of X are to be kept with their original values
corner	set to FALSE to not treat the last column specially. You may not specify both not and corner.
center	set to FALSE to not center columns of X first

Value

list with elements X, R, Rinv, xbar where xbar is the vector of means (vector of zeros if center=FALSE) @export

Index

cdf_dl, 3, *6*

func_link, 4
func_link_num, 4
func_V, 5

multipleDL, 5
multipleDL-package, 2

quantile_dl, *6*, 7

selectedQr, 8