Package: MDCcure (via r-universe)

October 29, 2026

Type Package

Title Martingale Dependence Tools and Testing for Mixture Cure Models

Version 0.1.0

Description Computes martingale difference correlation (MDC), martingale difference divergence, and their partial extensions to assess conditional mean dependence. The methods are based on Shao and Zhang (2014) <doi:10.1080/01621459.2014.887012>. Additionally, introduces a novel hypothesis test for evaluating covariate effects on the cure rate in mixture cure models, using MDC-based statistics. The methodology is described in Monroy-Castillo et al. (2025, manuscript submitted).

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

Suggests knitr, rmarkdown, pinp

LinkingTo Rcpp, RcppArmadillo, RcppParallel

Imports Rcpp, RcppParallel, ggplot2, ggtext, gridExtra, future, future.apply, smcure, npcure, survival

NeedsCompilation yes

SystemRequirements GNU make, TBB

URL https://github.com/CastleMon/MDCcure

BugReports https://github.com/CastleMon/MDCcure/issues

Author Blanca Monroy-Castillo [aut, cre], Amalia Jácome [aut], Ricardo Cao [aut], Ingrid Van Keilegom [aut], Ursula Müller [aut]

Maintainer Blanca Monroy-Castillo <blancamonroy.96@gmail.com>

Config/pak/sysreqs make

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

Date/Publication 2025-07-23 18:50:02 UTC

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

RemoteRef HEAD

RemoteSha a0b46dd8c426d0cc845e1f4ea65a0f1bd8ca61b1

Contents

goft 2
mdc 4
mdc_test 5
mdd 6
plotCure 7
pmdc 8
pmdd 9
testcov 10
testcov2 12

Index **15**

`goft` *Goodness-of-fit tests for the cure rate in a mixture cure model*

Description

The aim of this function is to test whether the cure rate p, as a function of the covariates, satisfies a certain parametric model.

Usage

```
goft(
  x,
  time,
  delta,
  model = c("logit", "probit", "cloglog"),
  theta0 = NULL,
  nsimb = 499,
  h = NULL
)
```

Arguments

`x`	A numeric vector representing the covariate of interest.
`time`	A numeric vector of observed survival times.
`delta`	A numeric vector indicating censoring status (1 = event occurred, 0 = censored).
`model`	A character string specifying the parametric model for the incidence part. Can be `"logit"`, `"probit"`, or `"cloglog"`.

`theta0`	Optional numeric vector with initial values for the model parameters. Default is `NULL`.
`nsimb`	An integer indicating the number of bootstrap replicates.Default is `499`.
`h`	Optional bandwidth value used for nonparametric estimation of the cure rate. Default is `NULL`.

Details

We want to test wether the cure rate p, as a function of covariates, satisfies a certain parametric model, such as, logistic, probit or cloglog model. The hypothesis are:

$$\mathcal{H}_0 : p = p_\theta \quad \text{for some} \quad \theta \in \Theta \quad \text{vs} \quad \mathcal{H}_1 : p \neq p_\theta \quad \text{for all} \quad \theta \in \Theta,$$

where Θ is a finite-dimensional parameter space and p_θ is a known function up to the parameter vector θ.

The test statistic is based on a weighted L_2 distance between a nonparametric estimator $\hat{p}(x)$ and a parametric estimator $p_{\hat{\theta}}(x)$ under $\mathcal{H}_0$, as proposed by Müller and Van Keilegom (2019):

$$\mathcal{T}_n = nh^{1/2} \int \left(\hat{p}(x) - p_{\hat{\theta}}(x)\right)^2 \pi(x) dx,$$

where $\pi(x)$ is a known weighting function, often chosen as the covariate density $f(x)$.

A practical empirical version of the statistic is given by:

$$\tilde{\mathcal{T}}_n = nh^{1/2} \frac{1}{n} \sum_{i=1}^{n} \left(\hat{p}(x_i) - p_{\hat{\theta}}(x_i)\right)^2,$$

where the integral is replaced by a sample average.

Value

A list with the following components:

statistic Numeric value of the test statistic.

p.value Numeric value of the bootstrap p-value for testing the null hypothesis.

bandwidth The bandwidth used.

References

Müller, U.U, & Van Keilegom, I. (2019). Goodness-of-fit tests for the cure rate in a mixture cure model. *Biometrika*, 106, 211-227. doi:10.1093/biomet/asy058

Examples

```
## Some artificial data
set.seed(123)
n <- 50
x <- runif(n, -2, 2) ## Covariate values
y <- rweibull(n, shape = .5*(x + 4)) ## True lifetimes
```

```
c <- rexp(n) ## Censoring values
p <- exp(2*x)/(1 + exp(2*x)) ## Probability of being susceptible
u <- runif(n)
t <- ifelse(u < p, pmin(y, c), c) ## Observed times
d <- ifelse(u < p, ifelse(y < c, 1, 0), 0) ## Uncensoring indicator
data <- data.frame(x = x, t = t, d = d)

goft(x, t, d, model = "logit")
```

---

mdc *Martingale Difference Correlation (MDC)*

Description

mdc computes the squared martingale difference correlation between a response variable Y and explanatory variable(s) X, measuring conditional mean dependence. X can be either univariate or multivariate.

Usage

```
mdc(X, Y, center = "U")
```

Arguments

X	A vector or matrix where rows represent samples and columns represent variables.
Y	A vector or matrix where rows represent samples and columns represent variables.
center	Character string indicating the centering method to use. One of: • "U": U-centering, which provides an unbiased estimator. • "D": Double-centering, which leads to a biased estimator.

Value

Returns the squared martingale difference correlation of Y given X.

References

Shao, X., and Zhang, J. (2014). Martingale difference correlation and its use in high-dimensional variable screening. *Journal of the American Statistical Association*, **109**(507), 1302-1318. doi:10.1080/01621459.2014.887012.

See Also

mdd, mdc_test

Examples

```
# Generate example data
set.seed(123)
n <- 50
x <- matrix(rnorm(n * 5), nrow = n)  # multivariate data with 5 variables
y <- rbinom(n, 1, 0.5)               # binary covariate

# Compute MDC with U-centering
mdc(x, y, center = "U")

# Compute MDC with double-centering
mdc(x, y, center = "D")
```

`mdc_test` *MDC-Based Dependence Tests Between Multivariate Data and a Covariate*

Description

Computes dependence between a multivariate dataset `x` and a univariate covariate `y` using different variants of the MDC (martingale difference correlation) test.

Usage

```
mdc_test(x, y, method, permutations = 999, parallel = TRUE, ncores = -1)
```

Arguments

`x`	Vector or matrix where rows represent samples, and columns represent variables.
`y`	Covariate vector.
`method`	Character string indicating the test to perform. One of: • `"MDCU"`: U-centering permutation test. • `"MDCV"`: Double-centering permutation test. • `"FMDCU"`: Fast asymptotic test with U-centering. • `"All"`: All of the above.
`permutations`	Number of permutations. Defaults to 999.
`parallel`	Logical. Whether to use parallel computing. Defaults to `TRUE`.
`ncores`	Number of threads for parallel computing (used only if `parallel = TRUE`).

Value

A list containing the test results and p-values.

References

Shao, X., and Zhang, J. (2014). Martingale difference correlation...

Examples

```
set.seed(123)
x <- matrix(rnorm(50 * 5), nrow = 50)
y <- rbinom(50, 1, 0.5)
mdc_test(x, y, method = "FMDCU")
```

`mdd` *Martingale Difference Divergence (MDD)*

Description

`mdd` computes the squared martingale difference divergence (MDD) between response variable(s) `Y` and explanatory variable(s) `X`, measuring conditional mean dependence.

Usage

```
mdd(X, Y, center = "U")
```

Arguments

`X`	A vector or matrix where rows represent samples and columns represent variables.
`Y`	A vector or matrix where rows represent samples and columns represent variables.
`center`	Character string indicating the centering method to use. One of: • `"U"`: U-centering, which provides an unbiased estimator. • `"D"`: Double-centering, which leads to a biased estimator. Default is `"U"`.

Value

Returns the squared Martingale Difference Divergence of `Y` given `X`.

References

Shao, X., and Zhang, J. (2014). Martingale difference correlation and its use in high-dimensional variable screening. *Journal of the American Statistical Association*, **109**(507), 1302-1318. doi:10.1080/01621459.2014.887012.

Examples

```
# Generate example data
set.seed(123)
n <- 50
x <- matrix(rnorm(n * 5), nrow = n)  # multivariate explanatory variables
y_vec <- rbinom(n, 1, 0.5)           # univariate response
y_mat <- matrix(rnorm(n * 2), nrow = n)  # multivariate response

# Compute MDD with vector Y and U-centering
mdd(x, y_vec, center = "U")

# Compute MDD with matrix Y and double-centering
mdd(x, y_mat, center = "D")
```

`plotCure` *Plot Cure Probability: A Comparison of Nonparametric and Parametric Estimation*

Description

This function generates a plot comparing nonparametric and parametric estimations of cure probability in a univariate setting. The nonparametric estimate is displayed with 95% confidence bands, while the parametric estimate is based on a logit, probit or complementary log-log link. An optional covariate density curve can be added as a secondary axis.

Usage

```
plotCure(
  x,
  time,
  delta,
  main.title = NULL,
  title.x = NULL,
  model = "logit",
  theta = NULL,
  legend.pos = "bottom",
  density = TRUE,
  hsmooth = 10,
  npoints = 100
)
```

Arguments

`x`	A numeric vector containing the covariate values.
`time`	A numeric vector representing the observed survival times.
`delta`	A binary vector indicating the event status (1 = event, 0 = censored).

main.title	Character string for the main title of the plot. If NULL, a default is used.
title.x	Character string for the x-axis label. If NULL, a default is used.
model	A character string indicating the assumed model. Options include "logit", "probit", and "cloglog". Defaults to "logit".
theta	A numeric vector of length 2, specifying the coefficients for the logistic model to generate the parametric estimate.
legend.pos	A character string indicating the position of the legend. Options include "bottom", "top", "left", "right", "none", etc.
density	Logical; if TRUE, adds a secondary y-axis with the covariate density curve.
hsmooth	Numeric. Smoothing bandwidth parameter (h) for the cure probability estimator.
npoints	Integer. Number of points at which the estimator is evaluated over the covariate range.

Details

The function estimates the cure probability nonparametrically using the probcure function and overlays it with a parametric estimate obtained from a logistic regression model. Confidence intervals (95%) are included for the nonparametric estimate. Optionally, the density of the covariate can be shown as a shaded area with a secondary y-axis.

Value

A ggplot object representing the cure probability plot.

See Also

probcure

---

pmdc *Partial Martingale Difference Correlation (pMDC)*

Description

pmdd measures conditional mean dependence of Y given X, adjusting for the dependence on Z.

Usage

```
pmdc(X, Y, Z)
```

Arguments

X	A vector or matrix where rows represent samples and columns represent variables.
Y	A vector or matrix where rows represent samples and columns represent variables.
Z	A vector or matrix where rows represent samples and columns represent variables.

Value

Returns the squared partial martingale difference correlation of Y given X, adjusting for the dependence on Z.

References

Park, T., Shao, X., and Yao, S. (2015). Partial martingale difference correlation. *Electronic Journal of Statistics*, **9**(1), 1492-1517. doi:10.1214/15EJS1047.

Examples

```
# Generate example data
set.seed(123)
n <- 50
x <- matrix(rnorm(n * 5), nrow = n)  # explanatory variables
y <- matrix(rnorm(n), nrow = n)      # response variable
z <- matrix(rnorm(n * 2), nrow = n)  # conditioning variables

# Compute partial MDD
pmdd(x, y, z)
```

---

pmdd *Partial Martingale Difference Divergence (pMDD)*

Description

pmdd measures conditional mean dependence of Y given X, adjusting for the dependence on Z.

Usage

```
pmdd(X, Y, Z)
```

Arguments

X	A vector or matrix where rows represent samples and columns represent variables.
Y	A vector or matrix where rows represent samples and columns represent variables.
Z	A vector or matrix where rows represent samples and columns represent variables.

Value

Returns the squared partial martingale difference divergence of Y given X, adjusting for the dependence on Z.

References

Park, T., Shao, X., and Yao, S. (2015). Partial martingale difference correlation. *Electronic Journal of Statistics*, **9**(1), 1492-1517. doi:10.1214/15EJS1047.

Examples

```
# Generate example data
set.seed(123)
n <- 50
x <- matrix(rnorm(n * 5), nrow = n)  # explanatory variables
y <- matrix(rnorm(n), nrow = n)      # response variable
z <- matrix(rnorm(n * 2), nrow = n)  # conditioning variables

# Compute partial MDD
pmdd(x, y, z)
```

---

testcov *Covariate Hypothesis Test of the Cure Probability based on Martingale Difference Correlation*

Description

Performs nonparametric hypothesis tests to evaluate the association between a covariate and the cure probability in mixture cure models. Several test statistics are supported, including martingale difference correlation (MDC)-based tests and an alternative GOFT test.

Usage

```
testcov(
  x,
  time,
  delta,
  h = NULL,
  method = "FMDCU",
  P = 999,
  parallel = TRUE,
  ncores = -1
)
```

Arguments

x	A numeric vector representing the covariate of interest.
time	A numeric vector of observed survival times.
delta	A binary vector indicating censoring status: 1 for event and 0 for censored.

h — Bandwidth parameter for kernel smoothing. Either a positive numeric value, NULL, or the character string "bootstrap". If NULL, an optimal bandwidth is selected automatically. If "bootstrap", the bandwidth is selected using the bootstrap method proposed by López-Cheda et al. (2016).

method — Character string specifying the test to perform. One of:

- "MDCU": Martingale Difference Correlation with U-centering.
- "MDCV": Martingale Difference Correlation with double-centering.
- "FMDCU": Fast approximation of MDC with U-centering.
- "GOFT": Goodness-of-fit test for the cure model.
- "All": All of the above tests.

Default is "FMDCU".

P — Integer. Number of permutations or bootstrap replications used to compute the null distribution of the test statistic. For methods "MDCU" or "MDCV", this is the number of permutations. For the "GOFT" method, it is the number of bootstrap replications. Defaults to 999.

parallel — Logical. If TRUE, parallel computing is used to speed up computations. Default is TRUE.

ncores — Integer. Number of cores to use for parallel computing. If NULL, it defaults to one less than the number of available cores.

Details

The function computes a statistic, based on the methodology proposed by Monroy-Castillo et al., to test whether a covariate $\boldsymbol{X}$ has an effect on the cure probability.

$$\mathcal{H}_0 : \mathbb{E}(\nu|\boldsymbol{X}) \equiv 1 - p \quad \text{a.s.} \quad \text{vs} \quad \mathcal{H}_1 : \mathbb{E}(\nu|\boldsymbol{X}) \not\equiv 1 - p \quad \text{a.s.}$$

The main problem is that the response variable (cure indicator ν) is partially observed due to censoring. This is addressed by estimating the cure indicator using the methodology of Amico et al. (2021). We define $\tau = \sup_x \tau(x)$, with $\tau(x) = \inf\{t : S_0(t|x) = 0\}$. We assume $\tau < \infty$ and that follow-up is long enough so that $\tau < \tau_{G(x)}$ for all x. Therefore, individuals with censored observed times greater than τ are considered cured ($\nu = 1$).

Four tests are proposed: three are based on the martingale difference correlation (MDC). For the MDCU and MDCV tests, the null distribution is approximated via a permutation procedure. To provide a faster alternative, a chi-squared approximation is implemented for the MDCU test statistic (FMDCU). Additionally, a modified version of the goodness-of-fit test proposed by Müller and Van Keilegom (2019) is included (GOFT). The test statistic is given by:

$$\widehat{\mathcal{T}}_n = nh^{1/2} \frac{1}{n} \sum_{i=1}^{n} \left\{ \hat{p}_h(X_i) - \hat{p} \right\}^2 ,$$

where $\hat{p}_h(X_i)$ denotes the nonparametric estimator of the cure probability under the alternative hypothesis, and $\hat{p}$ denotes the nonparametric estimator of the cure probability under the null hypothesis. The approximation of the critical value for the test is done using the bootstrap procedure given in Section 3 of Müller and Van Keilegom (2019).

Value

A list containing:

- `test_results`: A list with the results (e.g., test statistics and p-values) of the selected test(s).
- `nu_hat`: A numeric vector of estimated cure probabilities.

References

Amico, M, Van Keilegom, I. & Han, B. (2021). Assessing cure status prediction from survival data using receiver operating characteristic curves. *Biometrika*, 108(3), 727–740. doi:10.1093/biomet/asaa080

López-Cheda, A., Cao, R., Jácome, M. A., & Van Keilegom, I. (2016). Nonparametric incidence estimation and bootstrap bandwidth selection in mixture cure models. *Computational Statistics & Data Analysis*, 100, 490–502. doi:10.1016/j.csda.2016.04.006

Müller, U.U, & Van Keilegom, I. (2019). Goodness-of-fit tests for the cure rate in a mixture cure model. *Biometrika*, 106, 211-227. doi:10.1093/biomet/asy058

Shao, X., & Zhang, J. (2014). Martingale difference correlation and its use in high-dimensional variable screening. *Journal of the American Statistical Association*, 105, 144-165. doi:10.1080/01621459.2014.887012

See Also

`mdc`, `mdd`, `mdc_test`, `testcov2`

Examples

```
## Some artificial data
set.seed(123)
n <- 50
x <- runif(n, -2, 2) ## Covariate values
y <- rweibull(n, shape = .5*(x + 4)) ## True lifetimes
c <- rexp(n) ## Censoring values
p <- exp(2*x)/(1 + exp(2*x)) ## Probability of being susceptible
u <- runif(n)
t <- ifelse(u < p, pmin(y, c), c) ## Observed times
d <- ifelse(u < p, ifelse(y < c, 1, 0), 0) ## Uncensoring indicator
data <- data.frame(x = x, t = t, d = d)

testcov(x, t, d)
```

`testcov2` *Hypothesis test for association between covariate and cure indicator adjusted by a second covariate*

Description

Performs a permutation-based test assessing the association between a primary covariate (x) and the cure indicator, while adjusting for a secondary covariate (z). The test calculates the p-value via permutation using the partial martingale difference correlation.

Usage

```
testcov2(x, time, z, delta, P = 999, H = NULL)
```

Arguments

x	Numeric vector. The primary covariate whose association with the latent cure indicator is tested.
time	Numeric vector. Observed survival or censoring times.
z	Numeric vector. Secondary covariate for adjustment.
delta	Numeric vector. Censoring indicator (1 indicates event occurred, 0 indicates censored).
P	Integer. Number of permutations used to compute the permutation p-value. Default is 999.
H	Optional numeric. Bandwidth parameter (currently unused, reserved for future extensions).

Details

In order to test if the cure rate depends on the covariate $\boldsymbol{X}$ given it depends on the covariate $\boldsymbol{Z}$. The hypotheses are

$$\mathcal{H}_0 : \mathbb{E}(\nu|\boldsymbol{X}) \equiv 1 - p(\boldsymbol{X}) \quad \text{a.s.} \quad \text{vs} \quad \mathcal{H}_1 : \mathbb{E}(\nu|\boldsymbol{X}) \not\equiv 1 - p(\boldsymbol{X}) \quad \text{a.s.}$$

The proxy of the cure rate under the null hypothesis $\mathcal{H}_0$ is obtained by:

$$\mathbb{I}(T > \tau) + (1 - \delta)\mathbb{I}(T \leq \tau) \, \frac{1 - p(\boldsymbol{Z})}{1 - p(\boldsymbol{Z}) + p(\boldsymbol{Z}) S_0(T|\boldsymbol{X}, \boldsymbol{Z})}.$$

The statistic for testing the covariate hypothesis is based on partial martingale difference correlation and it is given by:

$$\text{pMDC}_n(\hat{\nu}_{\boldsymbol{H}}|\boldsymbol{X}, \boldsymbol{Z})^2.$$

The null distribution is approximated using a permutation test.

Value

List with components:

statistic Numeric. The test statistic value.

p.value Numeric. The permutation p-value assessing the null hypothesis of no association between x and the latent cure indicator, adjusting for z.

References

Park, T., Saho, X. & Yao, S. (2015). Partial martingale difference correlation. *Electronic Journal of Statistics*, 9, 1492–1517. doi:10.1214/15EJS1047

See Also

`pmdc` for the partial martingale difference correlation, `pmdd` for the partial martingale difference divergence, `testcov` for the test for one covariate.

Index

goft, 2

mdc, 4, *12*
mdc_test, *4*, 5, *12*
mdd, *4*, 6, *12*

plotCure, 7
pmdc, 8, *14*
pmdd, 9, *14*
probcure, *8*

testcov, 10, *14*
testcov2, 12, *12*