Package: sievePH (via r-universe)

October 6, 2026

Type Package

Title Sieve Analysis Methods for Proportional Hazards Models

Version 1.1

Date 2024-05-15

Description Implements a suite of semiparametric and nonparametric kernel-smoothed estimation and testing procedures for continuous mark-specific stratified hazard ratio (treatment/placebo) models in a randomized treatment efficacy trial with a time-to-event endpoint. Semiparametric methods, allowing multivariate marks, are described in Juraska M and Gilbert PB (2013), Mark-specific hazard ratio model with multivariate continuous marks: an application to vaccine efficacy. Biometrics 69(2):328-337 <doi:10.1111/biom.12016>, and in Juraska M and Gilbert PB (2016), Mark-specific hazard ratio model with missing multivariate marks. Lifetime Data Analysis 22(4):606-25 <doi:10.1007/s10985-015-9353-9>. Nonparametric kernel-smoothed methods, allowing univariate marks only, are described in Sun Y and Gilbert PB (2012), Estimation of stratified mark-specific proportional hazards models with missing marks. Scandinavian Journal of Statistics}, 39(1):34-52 <doi:10.1111/j.1467-9469.2011.00746.x>, and in Gilbert PB and Sun Y (2015), Inferences on relative failure rates in stratified mark-specific proportional hazards models with missing marks, with application to human immunodeficiency virus vaccine efficacy trials. Journal of the Royal Statistical Society Series C: Applied Statistics, 64(1):49-73 <doi:10.1111/rssc.12067>. Both semiparametric and nonparametric approaches consider two scenarios: (1) the mark is fully observed in all subjects who experience the event of interest, and (2) the mark is subject to missingness-at-random in subjects who experience the event of interest. For models with missing marks, estimators are implemented based on (i) inverse probability weighting (IPW) of complete cases (for the semiparametric framework), and (ii) augmentation of the IPW estimating functions by leveraging correlations between the

mark and auxiliary data to 'impute' the augmentation term for subjects with missing marks (for both the semiparametric and nonparametric framework). The augmented IPW estimators are doubly robust and recommended for use with incomplete mark data. The semiparametric methods make two key assumptions: (i) the time-to-event is assumed to be conditionally independent of the mark given treatment, and (ii) the weight function in the semiparametric density ratio/biased sampling model is assumed to be exponential. Diagnostic testing procedures for evaluating validity of both assumptions are implemented. Summary and plotting functions are provided for estimation and inferential results.

URL https://github.com/mjuraska/sievePH

BugReports https://github.com/mjuraska/sievePH/issues

License GPL-2

Encoding UTF-8

Imports graphics, stats, survival, ggplot2, ggpubr, scales, plyr, np

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.1

NeedsCompilation yes

Author Michal Juraska [aut, cre], Li Li [ctb], Stephanie Wu [ctb]

Maintainer Michal Juraska <mjuraska@fredhutch.org>

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

Date/Publication 2024-05-18 02:39:03 UTC

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

RemoteRef HEAD

RemoteSha 5bcddf10707f494c2237513b31d08c985081f74d

Contents

ggplot_sieve 3
kernel_sievePH 6
kernel_sievePHaipw 9
plot.summary.sievePH 14
sievePH 16
sievePHaipw 18
sievePHipw 20
summary.kernel_sievePH 23
summary.sievePH 26
testDensRatioGOF 28
testIndepTimeMark 30

Index **32**

ggplot_sieve *Plotting Univariate Mark-Specific Proportional Hazards Model Fits Using* ggplot

Description

ggplot-style plotting for univariate marks. Point and interval estimates of the mark-specific treatment effect parameter specified by component `contrast` in `summary.sievePH` or `summary.kernel_sievePH` are plotted, together with scatter and box plots of the observed mark values by treatment.

Usage

```
ggplot_sieve(
  x,
  mark = NULL,
  tx = NULL,
  xlim = NULL,
  ylim = NULL,
  xtickAt = NULL,
  xtickLab = NULL,
  ytickAt = NULL,
  ytickLab = NULL,
  tickLabSize = 14,
  xlab = NULL,
  ylab = NULL,
  axisLabSize = 15,
  title = NULL,
  titleSize = 16,
  subtitle = NULL,
  subtitleSize = 10,
  txLab = c("Placebo", "Treatment"),
  txLabSize = 5,
  legendLabSize = 12,
  legendPosition = c(0.96, 1.08),
  legendJustification = c(1, 1),
  estLineSize = 1.6,
  ciLineSize = 1.2,
  boxplotWidth = 0.8,
  jitterFactor = 0.1,
  jitterSeed = 0,
  pointColor = c("blue", "red3"),
  pointSize = 1.7,
  bottomPlotMargin = c(-0.5, 0.3, 0, 0),
  topPlotMargin = c(0, 0.3, -0.12, 1.83),
  plotHeights = c(0.33, 0.67)
)
```

Arguments

x	an object returned by summary.sievePH or summary.kernel_sievePH
mark	a numeric vector specifying a univariate continuous mark. For subjects with a right-censored time-to-event, the value(s) in mark should be set to NA.
tx	a numeric vector indicating the treatment group (1 if treatment, 0 if placebo)
xlim	a numeric vector of length 2 specifying the x-axis range (NULL by default)
ylim	a numeric vector of length 2 specifying the y-axis range (NULL by default)
xtickAt	a numeric vector specifying the position of x-axis tickmarks (NULL by default)
xtickLab	a numeric vector specifying labels for tickmarks listed in xtickAt. If NULL (default), the labels are determined by xtickAt.
ytickAt	a numeric vector specifying the position of y-axis tickmarks (NULL by default)
ytickLab	a numeric vector specifying labels for tickmarks listed in ytickAt. If NULL (default), the labels are determined by ytickAt.
tickLabSize	a numeric value specifying the font size of tickmark labels along both axes in the bottom panel (14 by default)
xlab	a character string specifying the x-axis label (NULL by default)
ylab	a character string specifying the y-axis label (NULL by default)
axisLabSize	a numeric value specifying the font size of both axis labels in the bottom panel (15 by default)
title	a character string specifying the plot title (NULL by default)
titleSize	a numeric value specifying the font size of the plot title (16 by default)
subtitle	a character string specifying the plot subtitle (NULL by default)
subtitleSize	a numeric value specifying the font size of the plot subtitle (10 by default)
txLab	a character vector of length 2 specifying the placebo and treatment labels (in this order). The default labels are placebo and treatment.
txLabSize	a numeric value specifying the font size of labels txLab (5 by default)
legendLabSize	a numeric value specifying the font size of legend labels in the bottom panel (11 by default)
legendPosition	a numeric vector of length 2 specifying the position of the legend in the bottom panel (c(0.96, 1.08) by default), passed on to argument legend.position in theme()
legendJustification	a numeric vector of length 2 specifying the justification of the legend in the bottom panel (c(1, 1) by default), passed on to argument legend.justification in theme()
estLineSize	a numeric value specifying the line width for the point estimate of the mark-specific treatment effect (1.6 by default)
ciLineSize	a numeric value specifying the line width for the confidence limits for the mark-specific treatment effect (1.2 by default)
boxplotWidth	a numeric value specifying the width of each box in the box plot (0.8) by default

jitterFactor	a numeric value specifying the amount of vertical jitter (0.1 by default)
jitterSeed	a numeric value setting the seed of R's random number generator for jitter in the scatter plot (0 by default)
pointColor	a character vector of length 2 color-coding the placebo and treatment group (in this order) in the scatter plot (c("blue", "red3") by default)
pointSize	a numeric value specifying the size of data points in the scatter plot (1.7 by default)
bottomPlotMargin	a numeric vector, using cm as the unit, passed on to argument plot.margin in theme() for the bottom panel (c(-0.5, 0.3, 0, 0) by default)
topPlotMargin	a numeric vector, using "lines" as the unit, passed on to argument plot.margin in theme() for the top panel (c(0, 0.3, -0.12, 1.83) by default)
plotHeights	a numeric vector specifying relative heights of the top and bottom panels (c(0.33, 0.67) by default) passed on to argument heights in ggpubr::ggarrange()

Value

A ggplot object.

See Also

plot.summary.sievePH, sievePH, summary.sievePH, kernel_sievePH, summary.kernel_sievePH

Examples

```
n <- 200
tx <- rep(0:1, each=n/2)
tm <- c(rexp(n/2, 0.2), rexp(n/2, 0.2 * exp(-0.4)))
cens <- runif(n, 0, 15)
eventTime <- pmin(tm, cens, 3)
eventInd <- as.numeric(tm <= pmin(cens, 3))
mark <- ifelse(eventInd==1, c(rbeta(n/2, 2, 5), rbeta(n/2, 2, 2)), NA)
markRng <- range(mark, na.rm=TRUE)

# fit a model with a univariate mark using the sievePH method
fit1 <- sievePH(eventTime, eventInd, mark, tx)
sfit1 <- summary(fit1, markGrid=seq(markRng[1], markRng[2], length.out=10))
print(ggplot_sieve(sfit1, mark, tx))

# fit a model with a univariate mark using the kernel_sievePH method
fit2 <- kernel_sievePH(eventTime, eventInd, mark, tx,
                       tau = 3, tband = 0.5, hband = 0.3, nvgrid = 20,
                       nboot = NULL)
sfit2 <- summary(fit2)
print(ggplot_sieve(sfit2, mark, tx))
```

kernel_sievePH *Nonparametric Kernel-Smoothed Stratified Mark-Specific Proportional Hazards Model with a Univariate Continuous Mark, Fully Observed in All Failures.*

Description

kernel_sievePH implements estimation and hypothesis testing method of Sun et al. (2009) for a mark-specific proportional hazards model. The methods allow separate baseline mark-specific hazard functions for different baseline subgroups.

Usage

```
kernel_sievePH(
  eventTime,
  eventInd,
  mark,
  tx,
  zcov = NULL,
  strata = NULL,
  formulaPH = ~tx,
  tau = NULL,
  tband = NULL,
  hband = NULL,
  nvgrid = 100,
  a = NULL,
  b = NULL,
  ntgrid = NULL,
  nboot = 500,
  seed = NULL,
  maxit = 6
)
```

Arguments

eventTime	a numeric vector specifying the observed right-censored event time.
eventInd	a numeric vector indicating the event of interest (1 if event, 0 if right-censored).
mark	a numeric vector specifying a univariate continuous mark. No missing values are permitted for subjects with eventInd = 1. For subjects with eventInd = 0, the value(s) in mark should be set to NA.
tx	a numeric vector indicating the treatment group (1 if treatment, 0 if placebo).
zcov	a data frame with one row per subject specifying possibly time-dependent covariate(s) (not including tx). If no covariate is used, zcov should be set to the default of NULL.
strata	a numeric vector specifying baseline strata (NULL by default). If specified, a separate mark-specific baseline hazard is assumed for each stratum.

`formulaPH` a one-sided formula object (on the right side of the ~ operator) specifying the linear predictor in the proportional hazards model. Available variables to be used in the formula include `tx` and variable(s) in `zcov`. By default, `formulaPH` is specified as `~ tx`.

`tau` a numeric value specifying the duration of study follow-up period. Failures beyond `tau` are treated right-censored. There needs to be at least 10% of subjects (as a rule of thumb) remaining uncensored by `tau` for the estimation to be stable. By default, `tau` is set as the maximum of `eventTime`.

`tband` a numeric value between 0 and `tau` specifying the bandwidth of the kernel smoothing function over time. By default, `tband` is set as (`tau`-min(`eventTime`))/5.

`hband` a numeric value between 0 and 1 specifying the bandwidth of the kernel smoothing function over mark. By default, `hband` is set as $4\sigma n^{-1/3}$ where σ is the estimated standard deviation of the observed marks for uncensored failure times and n is the number of subjects in the dataset. Larger bandwidths are recommended for higher percentages of missing marks.

`nvgrid` an integer value (100 by default) specifying the number of equally spaced mark values between the minimum and maximum of the observed mark for which the treatment effects are evaluated.

`a` a numeric value between the minimum and maximum of observed mark values specifying the lower bound of the range for testing the null hypotheses H_{10} : $HR(v) = 1$ and H_{20} : $HR(v)$ does not depend on v, for $v \in [a, b]$; By default, `a` is set as `(max(mark) - min(mark))/nvgrid + min(mark)`.

`b` a numeric value between the minimum and maximum of observed mark specifying the upper bound of the range for testing the null hypotheses $H_{10} : HR(v) = 1$ and $H_{20} : HR(v)$ does not depend on v, for $v \in [a, b]$; By default, `b` is set as $max(mark)$.

`ntgrid` an integer value (`NULL` by default) specifying the number of equally spaced time points for which the mark-specific baseline hazard functions are evaluated. If `NULL`, baseline hazard functions are not evaluated.

`nboot` number of bootstrap iterations (500 by default) for simulating the distributions of test statistics. If `NULL`, the hypotheses tests are not performed.

`seed` an integer specifying the random number generation seed for reproducing the test statistics and p-values. By default, a specific seed is not set.

`maxit` Maximum number of iterations to attempt for convergence in estimation. The default is 6.

Details

`kernel_sievePH` analyzes data from a randomized placebo-controlled trial that evaluates treatment efficacy for a time-to-event endpoint with a continuous mark. The parameter of interest is the ratio of the conditional mark-specific hazard functions (treatment/placebo), which is based on a stratified mark-specific proportional hazards model. This model assumes no parametric form for the baseline hazard function nor the treatment effect across different mark values.

Value

An object of class `kernel_sievePH` which can be processed by `summary.kernel_sievePH` to obtain or print a summary of the results. An object of class `kernel_sievePH` is a list containing the following components:

- `H10`: a data frame with test statistics (first row) and corresponding p-values (second row) for testing $H_{10} : HR(v) = 1$ for v $\in [a, b]$. Columns `TSUP1` and `Tint1` include test statistics and p-values for testing H_{10} vs. $H_{1a} : HR(v) \neq 1$ for any v $\in [a, b]$ (general alternative). Columns `TSUP1m` and `Tint1m` include test statistics and p-values for testing H_{10} vs. $H_{1m} : HR(v) \leq 1$ with strict inequality for some v in $[a, b]$ (monotone alternative). `TSUP1` and `TSUP1m` are based on extensions of the classic Kolmogorov-Smirnov supremum-based test. `Tint1` and `Tint1m` are based on generalizations of the integration-based Cramer-von Mises test. `Tint1` and `Tint1m` involve integration of deviations over the whole range of the mark. If `nboot` is `NULL`, `H10` is returned as `NULL`.
- `H20`: a data frame with test statistics (first row) and corresponding p-values (second row) for testing H_{20}: HR(v) does not depend on v $\in [a, b]$. Columns `TSUP2` and `Tint2` include test statistics and p-values for testing H_{20} vs. H_{2a}: HR depends on v $\in [a, b]$ (general alternative). Columns `TSUP2m` and `Tint2m` include test statistics and p-values for testing H_{20} vs. H_{2m}: HR increases as v increases $\in [a, b]$ (monotone alternative). `TSUP2` and `TSUP2m` are based on extensions of the classic Kolmogorov-Smirnov supremum-based test. `Tint2` and `Tint2m` are based on generalizations of the integration-based Cramer-von Mises test. `Tint2` and `Tint2m` involve integration of deviations over the whole range of the mark. If `nboot` is `NULL`, `H20` is returned as `NULL`.
- `estBeta`: a data frame summarizing point estimates and standard errors of the mark-specific coefficients for treatment at equally-spaced values between the minimum and the maximum of the observed mark values.
- `cBproc1`: a data frame containing equally-spaced mark values in the column `Mark`, test processes $Q^{(1)}(v)$ for observed data in the column `Observed`, and $Q^{(1)}(v)$ for `nboot` independent sets of normal samples in the columns S1, S2, $\cdots$. If `nboot` is `NULL`, `cBproc1` is returned as `NULL`.
- `cBproc2`: a data frame containing equally-spaced mark values in the column `Mark`, test processes $Q^{(2)}(v)$ for observed data in the column `Observed`, and $Q^{(2)}(v)$ for `nboot` independent sets of normal samples in the columns S1, S2, $\cdots$. If `nboot` is `NULL`, `cBproc2` is returned as `NULL`.
- `Lambda0`: an array of dimension K x nvgrid x ntgrid for the kernel-smoothed baseline hazard function $\lambda_{0k}, k = 1, \ldots, K$ where K is the number of strata. If `ntgrid` is `NULL` (by default), `Lambda0` is returned as `NULL`.

References

Sun, Y., Gilbert, P. B., & McKeague, I. W. (2009). Proportional hazards models with continuous marks. *Annals of statistics*, 37(1), 394.

Yang, G., Sun, Y., Qi, L., & Gilbert, P. B. (2017). Estimation of stratified mark-specific proportional hazards models under two-phase sampling with application to HIV vaccine efficacy trials. *Statistics in biosciences*, 9, 259-283.

Examples

```
set.seed(20240410)
beta <- 2.1
gamma <- -1.3
n <- 200
tx <- rep(0:1, each = n / 2)
tm <- c(rexp(n / 2, 0.2), rexp(n / 2, 0.2 * exp(gamma)))
cens <- runif(n, 0, 15)
eventTime <- pmin(tm, cens, 3)
eventInd <- as.numeric(tm <= pmin(cens, 3))
alpha <- function(b){ log((1 - exp(-2)) * (b - 2) / (2 * (exp(b - 2) - 1))) }
mark0 <- log(1 - (1 - exp(-2)) * runif(n / 2)) / (-2)
mark1 <- log(1 + (beta - 2) * (1 - exp(-2)) * runif(n / 2) / (2 * exp(alpha(beta)))) /
  (beta - 2)
mark <- ifelse(eventInd == 1, c(mark0, mark1), NA)
# the true TE(v) curve underlying the data-generating mechanism is:
# TE(v) = 1 - exp{alpha(beta) + beta * v + gamma}

# complete-case estimation discards rows with a missing mark
fit <- kernel_sievePH(eventTime, eventInd, mark, tx, tau = 3, tband = 0.5,
                      hband = 0.3, nvgrid = 20, nboot = 20)
```

---

kernel_sievePHaipw *Nonparametric Kernel-Smoothed Stratified Mark-Specific Proportional Hazards Model with a Univariate Continuous Mark, Missing-at-Random in Some Failures*

Description

`kernel_sievePH` implements estimation methods of Sun and Gilbert (2012) and hypothesis testing methods of Gilbert and Sun (2015) for a mark-specific proportional hazards model accommodating that some failures have a missing mark. The methods allow separate baseline mark-specific hazard functions for different baseline subgroups. Missing marks are handled via augmented IPW (AIPW) approach.

Usage

```
kernel_sievePHaipw(
  eventTime,
  eventInd,
  mark,
  tx,
  aux = NULL,
  auxType = NULL,
  zcov = NULL,
  strata = NULL,
  formulaPH = ~tx,
```

```
  formulaMiss = NULL,
  formulaAux = NULL,
  tau = NULL,
  tband = NULL,
  hband = NULL,
  nvgrid = 100,
  a = NULL,
  b = NULL,
  ntgrid = NULL,
  nboot = 500,
  seed = NULL,
  maxit = 6
)
```

Arguments

eventTime	a numeric vector specifying the observed right-censored event time.
eventInd	a numeric vector indicating the event of interest (1 if event, 0 if right-censored).
mark	a numeric vector specifying a univariate continuous mark subject to missingness at random. Missing mark values should be set to NA. For subjects with eventInd = 0, the value in mark should also be set to NA.
tx	a numeric vector indicating the treatment group (1 if treatment, 0 if placebo).
aux	a numeric vector specifying a binary or a continuous auxiliary covariate which may be potentially useful for predicting missingness, i.e, the probability of missing, and for informing about the distribution of missing marks. The mark missingness model only requires that the auxiliary covariates be observed in subjects who experienced the event of interest. For subjects with eventInd = 0, the value in aux may be set to NA. If no auxiliary covariate is used, set aux to the default of NULL.
auxType	a character string describing the data type of aux if aux is used. Data types allowed include "binary" and "continuous". If aux is not used, auxType should be set to the default of NULL.
zcov	a data frame with one row per subject specifying possibly time-dependent covariate(s) (not including tx). If no covariate is used, zcov should be set to the default of NULL.
strata	a numeric vector specifying baseline strata (NULL by default). If specified, a separate mark-specific baseline hazard is assumed for each stratum. It also allows the models of the probability of complete-case and of the mark distribution to differ across strata.
formulaPH	a one-sided formula object (on the right side of the ~ operator) specifying the linear predictor in the proportional hazards model. Available variables to be used in the formula include tx and variable(s) in zcov. By default, formulaPH is specified as ~ tx.
formulaMiss	a one-sided formula object (on the right side of the ~ operator) specifying the linear predictor in the logistic regression model used for predicting the probability of observing the mark. formulaMiss must be provided for the AIPW method.

Available variables to be used in the formula include `eventTime`, `tx`, `aux`, and variable(s) in `zcov`.

`formulaAux` a one-sided formula object (on the right side of the ~ operator) specifying the variables used for estimating the conditional distribution of `aux`. If `aux` is binary, the formula specifies the linear predictor in a logistic regression and if `aux` is continuous, the formula provides a symbolic description of variables used in kernel conditional density estimation. `formulaAux` is optional for the `AIPW` estimation procedure. Available variables to be used in the formula include `eventTime`, `tx`, `mark`, and variable(s) in `zcov`.

`tau` a numeric value specifying the duration of study follow-up period. Failures beyond `tau` are treated right-censored. There needs to be at least 10% of subjects (as a rule of thumb) remaining uncensored by `tau` for the estimation to be stable. By default, `tau` is set as the maximum of `eventTime`.

`tband` a numeric value between 0 and `tau` specifying the bandwidth of the kernel smoothing function over time. By default, `tband` is set as (`tau`-min(`eventTime`))/5.

`hband` a numeric value between 0 and 1 specifying the bandwidth of the kernel smoothing function over mark. By default, `hband` is set as $4\sigma n^{-1/3}$ where σ is the estimated standard deviation of the observed marks for uncensored failure times and n is the number of subjects in the dataset. Larger bandwidths are recommended for higher percentages of missing marks.

`nvgrid` an integer value (100 by default) specifying the number of equally spaced mark values between the minimum and maximum of the observed mark for which the treatment effects are evaluated.

`a` a numeric value between the minimum and maximum of observed mark values specifying the lower bound of the range for testing the null hypotheses H_{10} : $HR(v) = 1$ and H_{20} : $HR(v)$ does not depend on v, for $v \in [a, b]$; By default, `a` is set as `(max(mark) - min(mark))/nvgrid + min(mark)`.

`b` a numeric value between the minimum and maximum of observed mark specifying the upper bound of the range for testing the null hypotheses $H_{10} : HR(v) = 1$ and $H_{20} : HR(v)$ does not depend on v, for $v \in [a, b]$; By default, `b` is set as $max(mark)$.

`ntgrid` an integer value (`NULL` by default) specifying the number of equally spaced time points for which the mark-specific baseline hazard functions are evaluated. If `NULL`, baseline hazard functions are not evaluated.

`nboot` number of bootstrap iterations (500 by default) for simulating the distributions of test statistics. If `NULL`, the hypotheses tests are not performed.

`seed` an integer specifying the random number generation seed for reproducing the test statistics and p-values. By default, a specific seed is not set.

`maxit` Maximum number of iterations to attempt for convergence in estimation. The default is 6.

Details

`kernel_sievePH` analyzes data from a randomized placebo-controlled trial that evaluates treatment efficacy for a time-to-event endpoint with a continuous mark. The parameter of interest is the

ratio of the conditional mark-specific hazard functions (treatment/placebo), which is based on a stratified mark-specific proportional hazards model. This model assumes no parametric form for the baseline hazard function nor the treatment effect across different mark values. For data with missing marks, the estimation procedure leverages auxiliary predictors of whether the mark is observed and augments the IPW estimator with auxiliary predictors of the missing mark value.

Value

An object of class `kernel_sievePH` which can be processed by `summary.kernel_sievePH` to obtain or print a summary of the results. An object of class `kernel_sievePH` is a list containing the following components:

- `H10`: a data frame with test statistics (first row) and corresponding p-values (second row) for testing $H_{10} : HR(v) = 1$ for v $\in [a, b]$. Columns `TSUP1` and `Tint1` include test statistics and p-values for testing H_{10} vs. $H_{1a} : HR(v) \neq 1$ for any v $\in [a, b]$ (general alternative). Columns `TSUP1m` and `Tint1m` include test statistics and p-values for testing H_{10} vs. $H_{1m} : HR(v) \leq 1$ with strict inequality for some v in $[a, b]$ (monotone alternative). `TSUP1` and `TSUP1m` are based on extensions of the classic Kolmogorov-Smirnov supremum-based test. `Tint1` and `Tint1m` are based on generalizations of the integration-based Cramer-von Mises test. `Tint1` and `Tint1m` involve integration of deviations over the whole range of the mark. If `nboot` is `NULL`, `H10` is returned as `NULL`.
- `H20`: a data frame with test statistics (first row) and corresponding p-values (second row) for testing H_{20}: HR(v) does not depend on v $\in [a, b]$. Columns `TSUP2` and `Tint2` include test statistics and p-values for testing H_{20} vs. H_{2a}: HR depends on v $\in [a, b]$ (general alternative). Columns `TSUP2m` and `Tint2m` include test statistics and p-values for testing H_{20} vs. H_{2m}: HR increases as v increases $\in [a, b]$ (monotone alternative). `TSUP2` and `TSUP2m` are based on extensions of the classic Kolmogorov-Smirnov supremum-based test. `Tint2` and `Tint2m` are based on generalizations of the integration-based Cramer-von Mises test. `Tint2` and `Tint2m` involve integration of deviations over the whole range of the mark. If `nboot` is `NULL`, `H20` is returned as `NULL`.
- `estBeta`: a data frame summarizing point estimates and standard errors of the mark-specific coefficients for treatment at equally-spaced values between the minimum and the maximum of the observed mark values.
- `cBproc1`: a data frame containing equally-spaced mark values in the column `Mark`, test processes $Q^{(1)}(v)$ for observed data in the column `Observed`, and $Q^{(1)}(v)$ for `nboot` independent sets of normal samples in the columns S1, S2, $\cdots$. If `nboot` is `NULL`, `cBproc1` is returned as `NULL`.
- `cBproc2`: a data frame containing equally-spaced mark values in the column `Mark`, test processes $Q^{(2)}(v)$ for observed data in the column `Observed`, and $Q^{(2)}(v)$ for `nboot` independent sets of normal samples in the columns S1, S2, $\cdots$. If `nboot` is `NULL`, `cBproc2` is returned as `NULL`.
- `Lambda0`: an array of dimension K x nvgrid x ntgrid for the kernel-smoothed baseline hazard function $\lambda_{0k}, k = 1, \ldots, K$ where K is the number of strata. If `ntgrid` is `NULL` (by default), `Lambda0` is returned as `NULL`.

References

Gilbert, P. B. and Sun, Y. (2015). Inferences on relative failure rates in stratified mark-specific proportional hazards models with missing marks, with application to human immunodeficiency

virus vaccine efficacy trials. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 64(1), 49-73.

Sun, Y. and Gilbert, P. B. (2012). Estimation of stratified mark-specific proportional hazards models with missing marks. *Scandinavian Journal of Statistics*, 39(1), 34-52.

Yang, G., Sun, Y., Qi, L., & Gilbert, P. B. (2017). Estimation of stratified mark-specific proportional hazards models under two-phase sampling with application to HIV vaccine efficacy trials. *Statistics in biosciences*, 9, 259-283.

Examples

```
set.seed(20240410)
beta <- 2.1
gamma <- -1.3
n <- 200
tx <- rep(0:1, each = n / 2)
tm <- c(rexp(n / 2, 0.2), rexp(n / 2, 0.2 * exp(gamma)))
cens <- runif(n, 0, 15)
eventTime <- pmin(tm, cens, 3)
eventInd <- as.numeric(tm <= pmin(cens, 3))
alpha <- function(b){ log((1 - exp(-2)) * (b - 2) / (2 * (exp(b - 2) - 1))) }
mark0 <- log(1 - (1 - exp(-2)) * runif(n / 2)) / (-2)
mark1 <- log(1 + (beta - 2) * (1 - exp(-2)) * runif(n / 2) / (2 * exp(alpha(beta)))) /
  (beta - 2)
mark <- ifelse(eventInd == 1, c(mark0, mark1), NA)
# the true TE(v) curve underlying the data-generating mechanism is:
# TE(v) = 1 - exp{alpha(beta) + beta * v + gamma}

# a binary auxiliary covariate
A <- sapply(exp(-0.5 - 0.2 * mark) / (1 + exp(-0.5 - 0.2 * mark)),
            function(p){ ifelse(is.na(p), NA, rbinom(1, 1, p)) })
linPred <- 1 + 0.4 * tx - 0.2 * A
probs <- exp(linPred) / (1 + exp(linPred))
R <- rep(NA, n)
while (sum(R, na.rm = TRUE) < 10){
  R[eventInd == 1] <- sapply(probs[eventInd == 1],
                             function(p){ rbinom(1, 1, p) })
}
# a missing-at-random mark
mark[eventInd == 1] <- ifelse(R[eventInd == 1] == 1, mark[eventInd == 1], NA)

# AIPW estimation; auxiliary covariate is used (not required)
fit <- kernel_sievePHaipw(eventTime, eventInd, mark, tx, aux = A,
                          auxType = "binary", formulaMiss = ~ eventTime,
                          formulaAux = ~ eventTime + tx + mark,
                          tau = 3, tband = 0.5, hband = 0.3, nvgrid = 20,
                          nboot = 20)
```

plot.summary.sievePH *Plotting Mark-Specific Proportional Hazards Model Fits*

Description

plot method for class summary.sievePH and summary.kernel_sievePH. For univariate marks, it plots point and interval estimates of the mark-specific treatment effect parameter specified by contrast in summary.sievePH, and, optionally, scatter/box plots of the observed mark values by treatment. For bivariate marks, plotting is restricted to the point estimate, which is displayed as a surface. No plotting is provided for marks of higher dimensions.

Usage

```
## S3 method for class 'summary.sievePH'
plot(
  x,
  mark = NULL,
  tx = NULL,
  xlim = NULL,
  ylim = NULL,
  zlim = NULL,
  xtickAt = NULL,
  xtickLab = NULL,
  ytickAt = NULL,
  ytickLab = NULL,
  xlab = NULL,
  ylab = NULL,
  zlab = NULL,
  txLab = c("Placebo", "Treatment"),
  title = NULL,
  ...
)
```

Arguments

x	an object returned by summary.sievePH or summary.kernel_sievePH
mark	either a numeric vector specifying a univariate continuous mark or a data frame specifying a multivariate continuous mark. For subjects with a right-censored time-to-event, the value(s) in mark should be set to NA. For summary.kernel_sievePH, mark must be univariate.
tx	a numeric vector indicating the treatment group (1 if treatment, 0 if placebo)
xlim	a numeric vector of length 2 specifying the x-axis range (NULL by default)
ylim	a numeric vector of length 2 specifying the y-axis range (NULL by default)
zlim	a numeric vector of length 2 specifying the z-axis range in a 3-dimensional plot (NULL by default)

xtickAt	a numeric vector specifing the position of x-axis tickmarks (NULL by default)
xtickLab	a numeric vector specifying labels for tickmarks listed in xtickAt. If NULL (default), the labels are determined by xtickAt.
ytickAt	a numeric vector specifing the position of y-axis tickmarks (NULL by default)
ytickLab	a numeric vector specifying labels for tickmarks listed in ytickAt. If NULL (default), the labels are determined by ytickAt.
xlab	a character string specifying the x-axis label (NULL by default)
ylab	a character string specifying the y-axis label (NULL by default)
zlab	a character string specifying the z-axis label in a 3-dimensional plot (NULL by default)
txLab	a character vector of length 2 specifying the placebo and treatment labels (in this order). The default labels are placebo and treatment.
title	a character string specifying the plot title (NULL by default)
...	other arguments to be passed to plotting functions

Details

For bivariate marks, markGrid in summary.sievePH must have equally spaced values for each component.

Value

None. The function is called solely for plot generation.

See Also

sievePH, sievePHipw, sievePHaipw and summary.sievePH

Examples

```
n <- 500
tx <- rep(0:1, each=n/2)
tm <- c(rexp(n/2, 0.2), rexp(n/2, 0.2 * exp(-0.4)))
cens <- runif(n, 0, 15)
eventTime <- pmin(tm, cens, 3)
eventInd <- as.numeric(tm <= pmin(cens, 3))
mark <- ifelse(eventInd==1, c(rbeta(n/2, 2, 5), rbeta(n/2, 2, 2)), NA)
markRng <- range(mark, na.rm=TRUE)

# fit a model with a univariate mark
fit <- sievePH(eventTime, eventInd, mark, tx)
sfit <- summary(fit, markGrid=seq(markRng[1], markRng[2], length.out=10))
plot(sfit, mark, tx)
```

`sievePH` *Semiparametric Estimation of Coefficients in a Mark-Specific Proportional Hazards Model with a Multivariate Continuous Mark, Fully Observed in All Failures*

Description

`sievePH` implements the semiparametric estimation method of Juraska and Gilbert (2013) for the multivariate mark- specific hazard ratio in the competing risks failure time analysis framework. It employs (i) the semiparametric method of maximum profile likelihood estimation in the treatment-to-placebo mark density ratio model (Qin, 1998) and (ii) the ordinary method of maximum partial likelihood estimation of the overall log hazard ratio in the Cox model. `sievePH` requires that the multivariate mark data are fully observed in all failures.

Usage

```
sievePH(eventTime, eventInd, mark, tx, strata = NULL)
```

Arguments

`eventTime`	a numeric vector specifying the observed right-censored time to the event of interest
`eventInd`	a numeric vector indicating the event of interest (1 if event, 0 if right-censored)
`mark`	either a numeric vector specifying a univariate continuous mark or a data frame specifying a multivariate continuous mark. No missing values are permitted for subjects with `eventInd = 1`. For subjects with `eventInd = 0`, the value(s) in `mark` should be set to `NA`.
`tx`	a numeric vector indicating the treatment group (1 if treatment, 0 if placebo)
`strata`	a numeric vector specifying baseline strata (`NULL` by default). If specified, a stratified Cox model is fit for estimating the marginal hazard ratio (i.e., a separate baseline hazard is assumed for each stratum). No stratification is used in estimation of the mark density ratio.

Details

`sievePH` considers data from a randomized placebo-controlled treatment efficacy trial with a time-to-event endpoint. The parameter of interest, the mark-specific hazard ratio, is the ratio (treatment/placebo) of the conditional mark-specific hazard functions. It factors as the product of the mark density ratio (treatment/placebo) and the ordinary marginal hazard function ignoring mark data. The mark density ratio is estimated using the method of Qin (1998), while the marginal hazard ratio is estimated using `coxph()` in the `survival` package. Both estimators are consistent and asymptotically normal. The joint asymptotic distribution of the estimators is detailed in Juraska and Gilbert (2013).

Value

An object of class `sievePH` which can be processed by `summary.sievePH` to obtain or print a summary of the results. An object of class `sievePH` is a list containing the following components:

- `DRcoef`: a numeric vector of estimates of coefficients ϕ in the weight function $g(v, \phi)$ in the density ratio model
- `DRlambda`: an estimate of the Lagrange multiplier in the profile score functions for ϕ (that arises by profiling out the nuisance parameter)
- `DRconverged`: a logical value indicating whether the estimation procedure in the density ratio model converged
- `logHR`: an estimate of the marginal log hazard ratio from `coxph()` in the `survival` package
- `cov`: the estimated joint covariance matrix of `DRcoef` and `logHR`
- `coxphFit`: an object returned by the call of `coxph()`
- `nPlaEvents`: the number of events observed in the placebo group
- `nTxEvents`: the number of events observed in the treatment group
- `mark`: the input object
- `tx`: the input object

References

Juraska, M. and Gilbert, P. B. (2013), Mark-specific hazard ratio model with multivariate continuous marks: an application to vaccine efficacy. *Biometrics* 69(2):328–337.

Qin, J. (1998), Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika* 85, 619–630.

See Also

`summary.sievePH`, `plot.summary.sievePH`, `testIndepTimeMark` and `testDensRatioGOF`

Examples

```
n <- 500
tx <- rep(0:1, each=n/2)
tm <- c(rexp(n/2, 0.2), rexp(n/2, 0.2 * exp(-0.4)))
cens <- runif(n, 0, 15)
eventTime <- pmin(tm, cens, 3)
eventInd <- as.numeric(tm <= pmin(cens, 3))
mark1 <- ifelse(eventInd==1, c(rbeta(n/2, 2, 5), rbeta(n/2, 2, 2)), NA)
mark2 <- ifelse(eventInd==1, c(rbeta(n/2, 1, 3), rbeta(n/2, 5, 1)), NA)

# fit a model with a univariate mark
fit <- sievePH(eventTime, eventInd, mark1, tx)

# fit a model with a bivariate mark
fit <- sievePH(eventTime, eventInd, data.frame(mark1, mark2), tx)
```

`sievePHaipw` *Semiparametric Augmented Inverse Probability Weighted Complete-Case Estimation of Coefficients in a Mark-Specific Proportional Hazards Model with a Multivariate Continuous Mark, Missing-at-Random in Some Failures*

Description

`sievePHaipw` implements the semiparametric augmented inverse probability weighted (AIPW) complete-case estimation method of Juraska and Gilbert (2015) for the multivariate mark- specific hazard ratio, with the mark subject to missingness at random. It extends Juraska and Gilbert (2013) by (i) weighting complete cases (i.e., subjects with complete marks) by the inverse of their estimated probabilities given auxiliary covariates and/or treatment, and (ii) adding an augmentation term (the conditional expected profile score given auxiliary covariates and/or treatment) to the IPW estimating equations in the density ratio model for increased efficiency and robustness to mis-specification of the missingness model (Robins et al., 1994). The probabilities of observing the mark are estimated by fitting a logistic regression model with a user-specified linear predictor. The mean profile score vector (the augmentation term) in the density ratio model is estimated by fitting a linear regression model with a user-specified linear predictor. Coefficients in the treatment-to-placebo mark density ratio model (Qin, 1998) are estimated by solving the AIPW estimating equations. The ordinary method of maximum partial likelihood estimation is employed for estimating the overall log hazard ratio in the Cox model.

Usage

```
sievePHaipw(
  eventTime,
  eventInd,
  mark,
  tx,
  aux = NULL,
  strata = NULL,
  formulaMiss,
  formulaScore
)
```

Arguments

`eventTime`	a numeric vector specifying the observed right-censored event time
`eventInd`	a numeric vector indicating the event of interest (1 if event, 0 if right-censored)
`mark`	either a numeric vector specifying a univariate continuous mark or a data frame specifying a multivariate continuous mark subject to missingness at random. Missing mark values should be set to `NA`. For subjects with `eventInd = 0`, the value(s) in `mark` should also be set to `NA`.
`tx`	a numeric vector indicating the treatment group (1 if treatment, 0 if placebo)

`aux` a data frame specifying auxiliary covariates predictive of the probability of observing the mark. The mark missingness model only requires that the auxiliary covariates be observed in subjects who experienced the event of interest. For subjects with `eventInd = 0`, the value(s) in `aux` may be set to `NA`.

`strata` a numeric vector specifying baseline strata (`NULL` by default). If specified, a stratified Cox model is fit for estimating the marginal hazard ratio (i.e., a separate baseline hazard is assumed for each stratum). No stratification is used in estimation of the mark density ratio.

`formulaMiss` a one-sided formula object specifying (on the right side of the `~` operator) the linear predictor in the logistic regression model used for predicting the probability of observing the mark. All terms in the formula except `tx` must be evaluable in the data frame `aux`.

`formulaScore` a one-sided formula object specifying (on the right side of the `~` operator) the linear predictor in the linear regression model used for predicting the expected profile score vector (the augmentation term) in the AIPW estimating equations in the density ratio model. All terms in the formula except `tx` must be evaluable in the data frame `aux`.

Details

`sievePHaipw` considers data from a randomized placebo-controlled treatment efficacy trial with a time-to-event endpoint. The parameter of interest, the mark-specific hazard ratio, is the ratio (treatment/placebo) of the conditional mark-specific hazard functions. It factors as the product of the mark density ratio (treatment/placebo) and the ordinary marginal hazard function ignoring mark data. The mark density ratio is estimated using the AIPW complete-case estimation method, following Robins et al. (1994) and extending Qin (1998), and the marginal hazard ratio is estimated using `coxph()` in the `survival` package. The asymptotic properties of the AIPW complete-case estimator are detailed in Juraska and Gilbert (2015).

Value

An object of class `sievePH` which can be processed by `summary.sievePH` to obtain or print a summary of the results. An object of class `sievePH` is a list containing the following components:

- `DRcoef`: a numeric vector of estimates of coefficients ϕ in the weight function $g(v, \phi)$ in the density ratio model
- `DRlambda`: an estimate of the Lagrange multiplier in the profile score functions for ϕ (that arises by profiling out the nuisance parameter)
- `DRconverged`: a logical value indicating whether the estimation procedure in the density ratio model converged
- `logHR`: an estimate of the marginal log hazard ratio from `coxph()` in the `survival` package
- `cov`: the estimated joint covariance matrix of `DRcoef` and `logHR`
- `coxphFit`: an object returned by the call of `coxph()`
- `nPlaEvents`: the number of events observed in the placebo group
- `nTxEvents`: the number of events observed in the treatment group
- `mark`: the input object
- `tx`: the input object

References

Juraska, M., and Gilbert, P. B. (2015), Mark-specific hazard ratio model with missing multivariate marks. *Lifetime Data Analysis* 22(4): 606-25.

Juraska, M. and Gilbert, P. B. (2013), Mark-specific hazard ratio model with multivariate continuous marks: an application to vaccine efficacy. *Biometrics* 69(2):328-337.

Qin, J. (1998), Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika* 85, 619-630.

Robins, J. M., Rotnitzky, A., and Zhao, L. P. (1994), Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association* 89(427): 846-866.

See Also

summary.sievePH, plot.summary.sievePH, testIndepTimeMark and testDensRatioGOF

Examples

```
n <- 500
tx <- rep(0:1, each=n / 2)
tm <- c(rexp(n / 2, 0.2), rexp(n / 2, 0.2 * exp(-0.4)))
cens <- runif(n, 0, 15)
eventTime <- pmin(tm, cens, 3)
eventInd <- as.numeric(tm <= pmin(cens, 3))
mark1 <- ifelse(eventInd==1, c(rbeta(n / 2, 2, 5), rbeta(n / 2, 2, 2)), NA)
mark2 <- ifelse(eventInd==1, c(rbeta(n / 2, 1, 3), rbeta(n / 2, 5, 1)), NA)
# a continuous auxiliary covariate
A <- (mark1 + 0.4 * runif(n)) / 1.4
linPred <- -0.8 + 0.4 * tx + 0.8 * A
probs <- exp(linPred) / (1 + exp(linPred))
R <- rep(NA, length(probs))
while (sum(R, na.rm=TRUE) < 10){
  R[eventInd==1] <- sapply(probs[eventInd==1], function(p){ rbinom(1, 1, p) })
}
# produce missing-at-random marks
mark1[eventInd==1] <- ifelse(R[eventInd==1]==1, mark1[eventInd==1], NA)
mark2[eventInd==1] <- ifelse(R[eventInd==1]==1, mark2[eventInd==1], NA)

# fit a model with a bivariate mark
fit <- sievePHaipw(eventTime, eventInd, mark=data.frame(mark1, mark2), tx,
                aux=data.frame(A), formulaMiss= ~ tx * A, formulaScore= ~ tx * A + I(A^2))
```

`sievePHipw` *Semiparametric Inverse Probability Weighted Complete-Case Estimation of Coefficients in a Mark-Specific Proportional Hazards Model with a Multivariate Continuous Mark, Missing-at-Random in Some Failures*

Description

sievePHipw implements the semiparametric inverse probability weighted (IPW) complete-case estimation method of Juraska and Gilbert (2015) for the multivariate mark- specific hazard ratio, with the mark subject to missingness at random. It extends Juraska and Gilbert (2013) by weighting complete cases by the inverse of their estimated probabilities given auxiliary covariates and/or treatment. The probabilities are estimated by fitting a logistic regression model with a user-specified linear predictor. Coefficients in the treatment-to-placebo mark density ratio model (Qin, 1998) are estimated by solving the IPW estimating equations. The ordinary method of maximum partial likelihood estimation is employed for estimating the overall log hazard ratio in the Cox model.

Usage

```
sievePHipw(
  eventTime,
  eventInd,
  mark,
  tx,
  aux = NULL,
  strata = NULL,
  formulaMiss
)
```

Arguments

eventTime	a numeric vector specifying the observed right-censored event time
eventInd	a numeric vector indicating the event of interest (1 if event, 0 if right-censored)
mark	either a numeric vector specifying a univariate continuous mark or a data frame specifying a multivariate continuous mark subject to missingness at random. Missing mark values should be set to NA. For subjects with eventInd = 0, the value(s) in mark should also be set to NA.
tx	a numeric vector indicating the treatment group (1 if treatment, 0 if placebo)
aux	a data frame specifying auxiliary covariates predictive of the probability of observing the mark. The mark missingness model only requires that the auxiliary covariates be observed in subjects who experienced the event of interest. For subjects with eventInd = 0, the value(s) in aux may be set to NA.
strata	a numeric vector specifying baseline strata (NULL by default). If specified, a stratified Cox model is fit for estimating the marginal hazard ratio (i.e., a separate baseline hazard is assumed for each stratum). No stratification is used in estimation of the mark density ratio.
formulaMiss	a one-sided formula object specifying (on the right side of the ~ operator) the linear predictor in the logistic regression model used for predicting the probability of observing the mark. All terms in the formula except tx must be evaluable in the data frame aux.

Details

sievePHipw considers data from a randomized placebo-controlled treatment efficacy trial with a time-to-event endpoint. The parameter of interest, the mark-specific hazard ratio, is the ratio (treatment/placebo) of the conditional mark-specific hazard functions. It factors as the product of the mark density ratio (treatment/placebo) and the ordinary marginal hazard function ignoring mark data. The mark density ratio is estimated using the IPW complete-case estimation method, extending Qin (1998), and the marginal hazard ratio is estimated using coxph() in the survival package. The asymptotic properties of the IPW complete-case estimator are detailed in Juraska and Gilbert (2015).

Value

An object of class sievePH which can be processed by summary.sievePH to obtain or print a summary of the results. An object of class sievePH is a list containing the following components:

- DRcoef: a numeric vector of estimates of coefficients ϕ in the weight function $g(v, \phi)$ in the density ratio model
- DRlambda: an estimate of the Lagrange multiplier in the profile score functions for ϕ (that arises by profiling out the nuisance parameter)
- DRconverged: a logical value indicating whether the estimation procedure in the density ratio model converged
- logHR: an estimate of the marginal log hazard ratio from coxph() in the survival package
- cov: the estimated joint covariance matrix of DRcoef and logHR
- coxphFit: an object returned by the call of coxph()
- nPlaEvents: the number of events observed in the placebo group
- nTxEvents: the number of events observed in the treatment group
- mark: the input object
- tx: the input object

References

Juraska, M., and Gilbert, P. B. (2015), Mark-specific hazard ratio model with missing multivariate marks. *Lifetime Data Analysis* 22(4): 606-25.

Juraska, M. and Gilbert, P. B. (2013), Mark-specific hazard ratio model with multivariate continuous marks: an application to vaccine efficacy. *Biometrics* 69(2):328-337.

Qin, J. (1998), Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika* 85, 619-630.

See Also

summary.sievePH, plot.summary.sievePH, testIndepTimeMark and testDensRatioGOF

Examples

```
n <- 500
tx <- rep(0:1, each=n / 2)
tm <- c(rexp(n / 2, 0.2), rexp(n / 2, 0.2 * exp(-0.4)))
cens <- runif(n, 0, 15)
eventTime <- pmin(tm, cens, 3)
eventInd <- as.numeric(tm <= pmin(cens, 3))
mark1 <- ifelse(eventInd==1, c(rbeta(n / 2, 2, 5), rbeta(n / 2, 2, 2)), NA)
mark2 <- ifelse(eventInd==1, c(rbeta(n / 2, 1, 3), rbeta(n / 2, 5, 1)), NA)
# a continuous auxiliary covariate
A <- (mark1 + 0.4 * runif(n)) / 1.4
linPred <- -0.8 + 0.4 * tx + 0.8 * A
probs <- exp(linPred) / (1 + exp(linPred))
R <- rep(NA, length(probs))
while (sum(R, na.rm=TRUE) < 10){
  R[eventInd==1] <- sapply(probs[eventInd==1], function(p){ rbinom(1, 1, p) })
}
# produce missing-at-random marks
mark1[eventInd==1] <- ifelse(R[eventInd==1]==1, mark1[eventInd==1], NA)
mark2[eventInd==1] <- ifelse(R[eventInd==1]==1, mark2[eventInd==1], NA)

# fit a model with a bivariate mark
fit <- sievePHipw(eventTime, eventInd, mark=data.frame(mark1, mark2), tx,
                  aux=data.frame(A), formulaMiss= ~ tx * A)
```

---

summary.kernel_sievePH

Summarizing Nonparametric Kernel-Smoothed Stratified Mark-Specific Proportional Hazards Model Fits

---

Description

summary method for an object of class kernel_sievePH.

Usage

```
## S3 method for class 'kernel_sievePH'
summary(
  object,
  contrast = c("te", "hr", "loghr"),
  sieveAlternative = c("twoSided", "oneSided"),
  confLevel = 0.95,
  ...
)

## S3 method for class 'summary.kernel_sievePH'
print(x, digits = 4, ...)
```

Arguments

object	an object of class kernel_sievePH, a result of a call to kernel_sievePH and kernel_sievePHaipw.
contrast	a character string specifying the treatment effect parameter of interest. The default value is "te" (treatment efficacy); other options are "hr" (hazard ratio) and "loghr" (log hazard ratio).
sieveAlternative	a character string specifying the alternative hypothesis for the sieve tests, which can be either "twoSided" (default) or "oneSided".
confLevel	the confidence level (0.95 by default) of reported confidence intervals
...	further arguments passed to or from other methods
x	an object of class summary.kernel_sievePH, usually a result of a call to summary.kernel_sievePH
digits	the number of significant digits to use when printing (4 by default)

Details

print.summary.kernel_sievePH prints a formatted summary of results. Inference about coefficients in the kernel-smoothed mark-specific proportional hazards model is tabulated. Additionally, a summary is generated from the tests of two relevant null hypotheses: (1) $\{H_0 : HR(v) = 1$ for all $v\}$, and (2) $\{H_0 : HR(v) = HR$ for all $v\}$. For the tests of (2), sieveAlternative controls the choice of the alternative hypothesis.

Value

An object of class summary.kernel_sievePH, which is a list with the following components:

- estBeta: a data frame summarizing point estimates and standard errors of the mark-specific coefficients for treatment.
- HRunity.2sided: a data frame with test statistics (first row) and corresponding p-values (second row) for testing $H_{10} : HR(v) = 1$ vs. $H_{1a} : HR(v) \neq 1$ for any v $\in [a, b]$ (general alternative). TSUP1 is based on an extension of the classic Kolmogorov-Smirnov supremum-based test. Tint1 is a generalization of the integration-based Cramer-von Mises test.
- HRunity.1sided: a data frame with test statistics (first row) and corresponding p-values (second row) for testing $H_{10} : HR(v) = 1$ vs. $H_{1m} : HR(v) \leq 1$ with strict inequality for some v $\in [a, b]$ (monotone alternative). TSUP1m is based on an extension of the classic Kolmogorov-Smirnov supremum-based test. Tint1m is a generalization of the integration-based Cramer-von Mises test.
- HRconstant.2sided: a data frame with test statistics (first row) and corresponding p-values (second row) for testing H_{20}: HR(v) does not depend on v $\in [a, b]$ vs. H_{2a}: HR depends on v $\in [a, b]$ (general alternative). TSUP2 is based on an extension of the classic Kolmogorov-Smirnov supremum-based test. Tint2 is a generalization of the integration-based Cramer-von Mises test. This component is available if sieveAlternative="twoSided".
- HRconstant.1sided: a data frame with test statistics (first row) and corresponding p-values (second row) for testing H_{20}: HR(v) does not depend on v $\in [a, b]$ vs. H_{2m}: HR increases as v increases $\in [a, b]$ (monotone alternative). TSUP2m is based on an extension of the classic Kolmogorov-Smirnov supremum-based test. Tint2m is a generalization of the integration-based Cramer-von Mises test. This component is available if sieveAlternative="oneSided".

- `te`: a data frame summarizing point and interval estimates of the mark-specific treatment efficacy on the grid of mark values defined by `nvgrid` spanning from the minimum and maximum of the mark (available if `contrast="te"`). The confidence level is specified by `confLevel`.
- `hr`: a data frame summarizing point and interval estimates of the mark-specific hazard ratio on the grid of mark values defined by `nvgrid` spanning from the minimum and maximum of the mark (available if `contrast="hr"`). The confidence level is specified by `confLevel`.
- `loghr`: a data frame summarizing point and interval estimates of the mark-specific log hazard ratio on the grid of mark values defined by `nvgrid` spanning from the minimum and maximum of the mark (available if `contrast="loghr"`). The confidence level is specified by `confLevel`.

References

Gilbert, P. B. and Sun, Y. (2015). Inferences on relative failure rates in stratified mark-specific proportional hazards models with missing marks, with application to human immunodeficiency virus vaccine efficacy trials. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 64(1), 49-73.

Sun, Y. and Gilbert, P. B. (2012). Estimation of stratified mark-specific proportional hazards models with missing marks. *Scandinavian Journal of Statistics*, 39(1), 34-52.

See Also

`kernel_sievePH`

Examples

```
set.seed(20240410)
beta <- 2.1
gamma <- -1.3
n <- 200
tx <- rep(0:1, each = n / 2)
tm <- c(rexp(n / 2, 0.2), rexp(n / 2, 0.2 * exp(gamma)))
cens <- runif(n, 0, 15)
eventTime <- pmin(tm, cens, 3)
eventInd <- as.numeric(tm <= pmin(cens, 3))
alpha <- function(b){ log((1 - exp(-2)) * (b - 2) / (2 * (exp(b - 2) - 1))) }
mark0 <- log(1 - (1 - exp(-2)) * runif(n / 2)) / (-2)
mark1 <- log(1 + (beta - 2) * (1 - exp(-2)) * runif(n / 2) / (2 * exp(alpha(beta)))) /
  (beta - 2)
mark <- ifelse(eventInd == 1, c(mark0, mark1), NA)
# the true TE(v) curve underlying the data-generating mechanism is:
# TE(v) = 1 - exp{alpha(beta) + beta * v + gamma}

# a binary auxiliary covariate
A <- sapply(exp(-0.5 - 0.2 * mark) / (1 + exp(-0.5 - 0.2 * mark)),
            function(p){ ifelse(is.na(p), NA, rbinom(1, 1, p)) })
linPred <- 1 + 0.4 * tx - 0.2 * A
probs <- exp(linPred) / (1 + exp(linPred))
R <- rep(NA, n)
while (sum(R, na.rm = TRUE) < 10){
```

```
  R[eventInd == 1] <- sapply(probs[eventInd == 1],
                             function(p){ rbinom(1, 1, p) })
}
# a missing-at-random mark
mark[eventInd == 1] <- ifelse(R[eventInd == 1] == 1, mark[eventInd == 1], NA)

# AIPW estimation; auxiliary covariate is used (not required)
fit <- kernel_sievePHaipw(eventTime, eventInd, mark, tx, aux = A,
                          auxType = "binary", formulaMiss = ~ eventTime,
                          formulaAux = ~ eventTime + tx + mark,
                          tau = 3, tband = 0.5, hband = 0.3, nvgrid = 20,
                          nboot = 20)
sfit <- summary(fit)
# print the formatted summary
sfit
# treatment efficacy estimates on the grid
sfit$te
```

---

summary.sievePH *Summarizing Mark-Specific Proportional Hazards Model Fits*

Description

summary method for an object of class sievePH.

Usage

```
## S3 method for class 'sievePH'
summary(
  object,
  markGrid,
  contrast = c("te", "hr", "loghr"),
  sieveAlternative = c("twoSided", "oneSided"),
  confLevel = 0.95,
  ...
)

## S3 method for class 'summary.sievePH'
print(x, digits = 4, ...)
```

Arguments

object	an object of class sievePH, usually a result of a call to sievePH
markGrid	a matrix specifying a grid of multivariate mark values, where rows correspond to different values on the (multivariate) grid and columns correspond to components of the mark. A numeric vector is allowed for univariate marks. The point and interval estimates of the contrast are calculated on this grid.

contrast	a character string specifying the treatment effect parameter of interest. The default value is "te" (treatment efficacy); other options are "hr" (hazard ratio) and "loghr" (log hazard ratio).
sieveAlternative	a character string specifying the alternative hypothesis for the sieve tests, which can be either "twoSided" (default) or, in case of a univariate mark, "oneSided". The one-sided option is unavailable for a multivariate mark.
confLevel	the confidence level (0.95 by default) of reported confidence intervals
...	further arguments passed to or from other methods
x	an object of class summary.sievePH, usually a result of a call to summary.sievePH
digits	the number of significant digits to use when printing (4 by default)

Details

print.summary.sievePH prints a formatted summary of results. Inference about coefficients in the mark-specific proportional hazards model is tabulated. Additionally, a summary is generated from the likelihood-ratio and Wald tests of two relevant null hypotheses: (1) $\{H_0 : HR(v) = 1$ for all $v\}$, and (2) $\{H_0 : HR(v) = HR$ for all $v\}$. For the tests of (2) and a univariate mark, sieveAlternative controls the choice of the alternative hypothesis.

Value

An object of class summary.sievePH, which is a list with the following components:

- coef: a data frame summarizing point and interval estimates of the density ratio model coefficients and the marginal log hazard ratio (the confidence level is specified by confLevel), and p-values from the two-sided Wald test of the null hypothesis that the parameter equals zero
- pLR.HRunity.2sided: a numeric vector with two named components: pLR.dRatio.2sided is a p-value from the two-sided profile likelihood-ratio test of the null hypothesis $H_0 : \beta = 0$, where β is the vector of mark coefficients in the mark density ratio model, and pLR.cox.2sided is a p-value from the two-sided partial likelihood-ratio test of the null hypothesis $H_0 : \gamma = 0$, where γ is the marginal log hazard ratio in the Cox model. The two p-values are intended for the use of the Simes (1986) procedure as described on page 4 in Juraska and Gilbert (2013).
- pWald.HRunity.2sided: a p-value from the two-sided Wald test of the null hypothesis $\{H_0 : HR(v) = 1$ for all $v\}$
- pWtWald.HRunity.1sided: a p-value from the one-sided weighted Wald test of the null hypothesis $\{H_0 : HR(v) = 1$ for all $v\}$ against the alternative hypothesis $\{H_1 : HR < 1$ and $HR(v)$ is increasing in each component of $v\}$
- pLR.HRconstant.2sided: a p-value from the two-sided profile likelihood-ratio test of the null hypothesis $\{H_0 : HR(v) = HR$ for all $v\}$. This component is available if sieveAlternative="twoSided".
- pLR.HRconstant.1sided: a numeric vector with two named components: pLR.dRatio.2sided is a p-value from the two-sided profile likelihood-ratio test of the null hypothesis $\{H_0 : HR(v) = HR$ for all $v\}$, and estBeta is the point estimate of the univariate mark coefficient in the density ratio model. This component is available if the mark is univariate and sieveAlternative="oneSided".
- pWald.HRconstant.2sided: a p-value from the two-sided Wald test of the null hypothesis $\{H_0 : HR(v) = HR$ for all $v\}$. This component is available if sieveAlternative="twoSided".

- pWald.HRconstant.1sided: a p-value from the one-sided Wald test of the null hypothesis $\{H_0 : HR(v) = HR \text{ for all } v\}$ against the alternative hypothesis $\{H_1 : HR(v)$ is increasing in $v\}$. This component is available if the mark is univariate and sieveAlternative="oneSided".
- te: a data frame summarizing point and interval estimates of the mark-specific treatment efficacy on the grid of mark values in markGrid (available if contrast="te"). The confidence level is specified by confLevel.
- hr: a data frame summarizing point and interval estimates of the mark-specific hazard ratio on the grid of mark values in markGrid (available if contrast="hr"). The confidence level is specified by confLevel.
- loghr: a data frame summarizing point and interval estimates of the mark-specific log hazard ratio on the grid of mark values in markGrid (available if contrast="loghr"). The confidence level is specified by confLevel.

References

Juraska, M. and Gilbert, P. B. (2013), Mark-specific hazard ratio model with multivariate continuous marks: an application to vaccine efficacy. *Biometrics* 69(2):328–337.

See Also

sievePH

Examples

```
n <- 500
tx <- rep(0:1, each=n/2)
tm <- c(rexp(n/2, 0.2), rexp(n/2, 0.2 * exp(-0.4)))
cens <- runif(n, 0, 15)
eventTime <- pmin(tm, cens, 3)
eventInd <- as.numeric(tm <= pmin(cens, 3))
mark1 <- ifelse(eventInd==1, c(rbeta(n/2, 2, 5), rbeta(n/2, 2, 2)), NA)
mark2 <- ifelse(eventInd==1, c(rbeta(n/2, 1, 3), rbeta(n/2, 5, 1)), NA)

# fit a model with a bivariate mark
fit <- sievePH(eventTime, eventInd, data.frame(mark1, mark2), tx)
sfit <- summary(fit, markGrid=matrix(c(0.3, 0.3, 0.6, 0.3, 0.3, 0.6, 0.6, 0.6),
                                     ncol=2, byrow=TRUE))
# print the formatted summary
sfit
# treatment efficacy estimates on the grid
sfit$te
```

---

testDensRatioGOF *Goodness-of-Fit Test of the Validity of a Univariate or Multivariate Mark Density Ratio Model*

Description

testDensRatioGoF implements the complete-case goodness-of-fit test of Qin and Zhang (1997) for evaluating the validity of the specified mark density ratio model used for modeling a component of the mark-specific hazard ratio model in Juraska and Gilbert (2013). Multivariate marks are accommodated. Subjects who experienced the event of interest but their mark is missing are discarded.

Usage

```
testDensRatioGOF(
  eventInd,
  mark,
  tx,
  DRcoef = NULL,
  DRlambda = NULL,
  iter = 1000
)
```

Arguments

eventInd	a numeric vector indicating the event of interest (1 if event, 0 if right-censored)
mark	either a numeric vector specifying a univariate continuous mark or a data frame specifying a multivariate continuous mark. For subjects with a right-censored time-to-event, the value(s) in mark should be set to NA.
tx	a numeric vector indicating the treatment group (1 if treatment, 0 if placebo)
DRcoef	a numeric vector of the coefficients ϕ in the weight function $g(v, \phi) = \exp\{\phi^T(1, v)\}$ in the density ratio model. If NULL (default), the maximum profile likelihood estimates (Qin, 1998) of the coefficients are computed.
DRlambda	the Lagrange multiplier in the profile score functions for ϕ (that arises by profiling out the nuisance parameter). If NULL (default), the maximum profile likelihood estimate (Qin, 1998) of the Lagrange multiplier is computed.
iter	the number of bootstrap iterations (1000 by default)

Details

testDensRatioGoF performs a goodness-of-fit test for the exponential form of the weight function, i.e., $g(v, \phi) = \exp\{\phi^T(1, v)\}$. Other weight functions are not considered.

Value

Returns a list containing the following components:

- teststat: the value of the Kolmogorov-Smirnov-type test statistic
- pval: the bootstrap p-value from the Kolmogorov-Smirnov-type test of validity of the mark density ratio model
- DRcoef: the input object if different from NULL or a numeric vector of estimates of coefficients ϕ in the weight function $g(v, \phi)$ in the density ratio model

- DRlambda: the input object if different from NULL or an estimate of the Lagrange multiplier in the profile score functions for ϕ

References

Qin, J., & Zhang, B. (1997). A goodness-of-fit test for logistic regression models based on case-control data. *Biometrika*, 84(3), 609-618.

Juraska, M. and Gilbert, P. B. (2013), Mark-specific hazard ratio model with multivariate continuous marks: an application to vaccine efficacy. *Biometrics* 69(2):328-337.

Qin, J. (1998), Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika* 85, 619-630.

Examples

```
n <- 500
tx <- rep(0:1, each=n/2)
tm <- c(rexp(n/2, 0.2), rexp(n/2, 0.2 * exp(-0.4)))
cens <- runif(n, 0, 15)
eventTime <- pmin(tm, cens, 3)
eventInd <- as.numeric(tm <= pmin(cens, 3))
mark1 <- ifelse(eventInd==1, c(rbeta(n/2, 2, 5), rbeta(n/2, 2, 2)), NA)
mark2 <- ifelse(eventInd==1, c(rbeta(n/2, 1, 3), rbeta(n/2, 5, 1)), NA)

# test goodness-of-fit for a univariate mark
testDensRatioGOF(eventInd, mark1, tx, iter=15)

# test goodness-of-fit for a bivariate mark
testDensRatioGOF(eventInd, data.frame(mark1, mark2), tx, iter=15)
```

---

testIndepTimeMark *Kolmogorov-Smirnov-Type Test of Conditional Independence between the Time-to-Event and a Multivariate Mark Given Treatment*

Description

A nonparametric Komogorov-Smirnov-type test of the null hypothesis that the time-to-event T and a possibly multivariate mark V are conditionally independent given treatment Z as described in Juraska and Gilbert (2013). The conditional independence is a necessary assumption for parameter identifiability in the time-independent density ratio model. A bootstrap algorithm is used to compute the p-value.

Usage

```
testIndepTimeMark(data, iter = 1000)
```

Arguments

data	a data frame restricted to subjects in a given treatment group with the following columns (in this order): the observed right-censored time to the event of interest, the event indicator (1 if event, 0 if right-censored), and the mark variable (one column for each component, if multivariate)
iter	the number of bootstrap iterations (1000 by default) used for computing the p-value

Details

The test statistic is the supremum of the difference between the estimated conditional joint cumulative distribution function (cdf) of (T, V) given Z and the product of the estimated conditional cdfs of T and V given Z. The joint cdf is estimated by the nonparametric maximum likelihood estimator developed by Huang and Louis (1998). The marginal cdf of T is estimated as one minus the Kaplan-Meier estimator for the conditional survival function of T, and the cdf of V is estimated as the empirical cdf of the observed values of V. A bootstrap algorithm is used to compute the p-value.

Value

Returns the bootstrap p-value from the test of conditional independence between T and V given Z.

References

Juraska, M. and Gilbert, P. B. (2013), Mark-specific hazard ratio model with multivariate continuous marks: an application to vaccine efficacy. *Biometrics* 69(2):328–337.

Huang, Y. and Louis, T. (1998), Nonparametric estimation of the joint distribution of survival time and mark variables. *Biometrika* 85, 785–798.

Examples

```
n <- 500
tx <- rep(0:1, each=n/2)
tm <- c(rexp(n/2, 0.2), rexp(n/2, 0.2 * exp(-0.4)))
cens <- runif(n, 0, 15)
eventTime <- pmin(tm, cens, 3)
eventInd <- as.numeric(tm <= pmin(cens, 3))
mark1 <- ifelse(eventInd==1, c(rbeta(n/2, 2, 5), rbeta(n/2, 2, 2)), NA)
mark2 <- ifelse(eventInd==1, c(rbeta(n/2, 1, 3), rbeta(n/2, 5, 1)), NA)

# perform the test for a univariate mark in the placebo group
testIndepTimeMark(data.frame(eventTime, eventInd, mark1)[tx==0, ], iter=20)

# perform the test for a bivariate mark in the placebo group
testIndepTimeMark(data.frame(eventTime, eventInd, mark1, mark2)[tx==0, ], iter=20)
```

Index

ggplot_sieve, 3

kernel_sievePH, *5*, 6, *24, 25*
kernel_sievePHaipw, 9, *24*

plot.summary.sievePH, *5*, 14, *17*, *20*, *22*
print.summary.kernel_sievePH
(summary.kernel_sievePH), 23
print.summary.sievePH
(summary.sievePH), 26

sievePH, *5*, *15*, 16, *26*, *28*
sievePHaipw, *15*, 18
sievePHipw, *15*, 20
summary.kernel_sievePH, *3–5*, *8*, *12*, *14*, 23
summary.sievePH, *3–5*, *14, 15*, *17*, *19, 20*, *22*, 26

testDensRatioGOF, *17*, *20*, *22*, 28
testIndepTimeMark, *17*, *20*, *22*, 30