Package 'sievePH'

February 3, 2023

Type Package

Title Sieve Analysis Methods for Proportional Hazards Models

Version 1.0.4

Date 2023-02-03

Description Implements semiparametric estimation and testing procedures for a continuous, possibly multivariate, mark-specific hazard ratio (treatment/placebo) of an event of interest in a randomized treatment efficacy trial with a time-to-event endpoint, as 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>. The former considers continuous multivariate marks fully observed in all subjects who experience the event of interest, whereas the latter extends the previous work to allow multivariate marks that are subject to missingness-at-random. For models with missing marks, two estimators are implemented based on (i) inverse probability weighting (IPW) of complete cases, and (ii) augmentation of the IPW estimating functions by leveraging correlations between the mark and auxiliary data to 'impute' the expected profile score vectors for subjects with missing marks. The augmented IPW estimator is doubly robust and recommended for use with incomplete mark data. The 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

RoxygenNote 7.2.1

NeedsCompilation no

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

Maintainer Michal Juraska <mjuraska@fredhutch.org>

Repository CRAN

Date/Publication 2023-02-03 18:40:02 UTC

R topics documented:

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` 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`
`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` 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))
print(ggplot_sieve(sfit, mark, tx))
```

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

Description

`plot` method for class `summary.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`

`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)

`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.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 $\{H_0 : HR(v) = HR$ 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`.
- `te`: 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=20)

# test goodness-of-fit for a bivariate mark
testDensRatioGOF(eventInd, data.frame(mark1, mark2), tx, iter=20)
```

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