Package 'survPresmooth'

October 6, 2021

Type Package

Title Presmoothed Estimation in Survival Analysis

Version 1.1-11

Date 2021-10-06

Maintainer Ignacio Lopez de Ullibarri <ilu@udc.es>

Depends R(>= 3.0.1)

Description Presmoothed estimators of survival, density, cumulative and non-cumulative hazard functions with right-censored survival data. For details, see Lopez-de-Ullibarri and Jacome (2013) <doi:10.18637/jss.v054.i11>.

License GPL (>= 2)

NeedsCompilation yes

LazyData yes

Author Ignacio Lopez de Ullibarri [aut, cre],
Maria Amalia Jacome [aut]

Repository CRAN

Date/Publication 2021-10-06 11:50:02 UTC

R topics documented:

`control.presmooth` *Control Values for the presmooth() Function*

Description

This function returns a list of values for control parameters of the `presmooth` function.

Usage

```
control.presmooth(n.boot = c(5000, 1000), q.weight = c(0.2, 0.8),
   k = 1, length.grid.bw.plugin = 100, length.grid.ise = 100,
   pilot.par.ini = NULL, save.data = FALSE, save.mise = FALSE,
   na.action = na.omit)
```

Arguments

`n.boot`	A numeric vector of length 2 or, alternatively, a numeric value specifying the number of bootstrap resamples used in bootstrap bandwidth selection. If a vector, the first element is used for S and H estimation, and the second for f and h. The default is c(5000, 1000).
`q.weight`	A numeric vector of length 2 specifying the order of quantiles of the observed times that determine the left- and right-ends of the support of the weight function. The default is c(0.2, 0.8).
`k`	A numeric value specifying the multiple of the data range used as the maximum possible value for the selected plug-in or bootstrap bandwidth. The default is 1.
`length.grid.bw.plugin`	An integer specifying the number of points of the grid used for numerical integration (Simpson's rule) of integrals involved in plug-in bandwidth selection. The default is 100.
`length.grid.ise`	An integer specifying the number of points of the grid used for computing the integrated squared error by numerical integration (Simpson's rule) in bootstrap bandwidth selection. The default is 100.
`pilot.par.ini`	A numeric vector of length 8 specifying the initial values of optimization routines used for pilot bandwidth computation in plug-in and bootstrap bandwidth selection (although, in the last case, it is not needed for survival and cumulative hazard function estimation). If `NULL`, the default, values are internally computed.
`save.data`	A logical value specifying if the data must be saved as a component of the value returned by the `presmooth` function. The default is `FALSE`.
`save.mise`	A logical value specifying if the MISE computed for bootstrap bandwidth selection must be saved as a component of the value returned by the `presmooth` function. The default is `FALSE`.
`na.action`	A function specifying how to handle missing values. The default value, na.omit, deletes the rows containing missing values in the (internal) data frame formed by the `times` and `status` arguments of the call to the `presmooth` function.

Details

The default values of `n.boot` represent a trade-off between computational speed and precision in bootstrap bandwidth selection. If enough computing power or time are available, it is recommended to increase the number of bootstrap resamples up to 10000 (which could be very slow for hazard and density estimation).

The six first values of `pilot.par.ini` are three pairs of parameters specifying three Weibull distributions (in each pair, the first element is the shape parameter; the second one, the scale parameter). These three Weibull distributions form a mixture, the weights of its first two components being given by the last two values of `pilot.par.ini`.

Value

A list whose components are the arguments of the function, its defaults being replaced with the values the function was called with.

Author(s)

Ignacio Lopez-de-Ullibarri [aut, cre], Maria Amalia Jacome [aut]

See Also

`presmooth`

`presmooth` *Compute Presmoothed Estimators*

Description

This function computes presmoothed estimators of the main functions used in survival analysis (survival function, cumulative hazard function, density function and non-cumulative hazard function) with right-censored data.

Usage

```
presmooth(times, status, dataset = NULL, estimand = c("S", "H",
"f", "h"), bw.selec = c("fixed", "plug-in", "bootstrap"), presmoothing =
TRUE, fixed.bw = NULL, grid.bw.pil = NULL, grid.bw = NULL, kernel =
c("biweight", "triweight"), bound = c("none", "left","right", "both"),
x.est = NULL, control = control.presmooth())
```

Arguments

`times`	An object of mode numeric giving the observed, possibly right-censored times. If dataset is not `NULL` it is interpreted as the name of the corresponding variable of the dataset.

`status` An object of mode numeric giving the censoring status of the times coded in the `times` object. Censored values must be coded as 0, uncensored values as 1. If `dataset` is not `NULL` it is interpreted as the name of the corresponding variable of the dataset.

`dataset` An optional data frame in which the variables named in `times` and `status` are interpreted. If it is `NULL`, `times` and `status` must be objects of the workspace.

`estimand` An optional character string identifying the function to estimate: "S" for survival function, "H" for cumulative hazard function, "f" for density function and "h" for non-cumulative hazard function. Defaults to "S".

`bw.selec` An optional (partially matched) character string specifying the method of bandwidth selection. "fixed" if no bandwidth selection is done, in which case the bandwidth(s) given by the `fixed.bw` argument is (are) used, "plug-in" for plug-in bandwidth selection and "bootstrap" for bootstrap bandwidth selection. Defaults to "fixed".

`presmoothing` An optional logical value. If `TRUE`, the default, a presmoothed estimate is computed; if `FALSE`, the non-presmoothed counterpart (also including bandwidth selection) is computed.

`fixed.bw` An optional numeric vector with the fixed bandwidth(s) used when the value of the `bw.selec` argument is "fixed". It must be of length 1 for estimating survival and cumulative hazard functions, and of length 2 for density and hazard functions (in this case, the first element is the presmoothing bandwidth).

`grid.bw.pil` An optional numeric vector specifying the grid where the pilot bandwidth for the Nadaraya-Watson estimate of the conditional probability of uncensoring, p, will be selected. Not used in plug-in bandwidth selection for survival or cumulative hazard function estimation. If `NULL`, it is internally computed.

`grid.bw` An optional list of length 1 (for presmoothed estimation of survival and cumulative hazard functions, and non-presmoothed estimation of density and hazard functions) or 2 (for presmoothed estimation of density and hazard functions) whose component(s) (is) are (a) numeric vector(s) specifying the grid of bandwidths needed for bootstrap bandwith selection when the value of the `bw.selec` argument is "bootstrap". If it is a list of length 1, it can also be inputted as a numeric vector. If `NULL`, it is internally computed.

`kernel` A (partially matched) character string specifying the kernel function used. One of "biweight", for biweight kernel, and "triweight", for triweight kernel. Defaults to "biweight".

`bound` A (partially matched) character string specifying the end(s) of the data range where boundary-effect correction is applied. If "none", the default, no correction is done; if "left", "right" or "both", the correction is applied at the left, right or both ends, respectively.

`x.est` A numeric vector specifying the points where the estimate will be computed. Only meaningful for density and hazard function estimation. Internally computed when `NULL`, the default.

`control` A list of control values. The value returned by the `control.presmooth` function called without arguments is the default.

Details

In survival analysis with right-censored data, presmoothing (see references below for details) provides a method to obtain new estimators from classical estimators, essentially by replacing the indicator of non-censoring with a nonparametric estimate (in our implementation, through the use of Nadaraya-Watson regression estimator) of the conditional probability of uncensoring. The `presmooth` function computes presmoothed versions of: 1) Kaplan-Meier survival function estimator, 2) Nelson-Aalen cumulative hazard function estimator, 3) the kernel density function estimator of Foldes-Rejto-Winter, and 4) the kernel hazard function estimator of Tanner-Wong (similar to that proposed by Yandell and Ramlau-Hansen). All presmoothed estimators have at least one presmoothing bandwidth (the smoothing parameter of the Nadaraya-Watson estimator), but in the case of the kernel estimators of density and hazard they have an additional smoothing bandwidth scaling the kernel.

The `presmooth` function incorporates plug-in and bootstrap global bandwidth selectors for every estimator implemented. Plug-in bandwith selection is done according to Cao et al. (2005) in the case of survival and cumulative hazard estimation, and following Jacome et al. (2008) together with the results given in Cao and Lopez-de-Ullibarri (2007) in the case of density and hazard estimation. As for bootstrap bandwidth selection, our method follows the approach of Gonzalez-Manteiga et al. (1996). See Jacome et al. (2008) for more details in the case of density estimation. The weight function needed for the bootstrap bandwidth is taken as constantly equal to 1. The left and right endpoints of its support are fixed via the `q.weight` argument of the `control.presmooth` function (see the online help for this function) and form the `q.weight` component of the value returned by `presmooth`. An upper bound equal to the range of the observed times is set for the selected (plug-in or bootstrap) bandwidth. On the other hand, bandwidths can also be fixed by the user. When the presmoothing bandwidth is fixed at 0, the classical, non-presmoothed versions of the estimators are computed. Non-presmoothed estimates are also obtained by calling `presmooth` with the value of the `presmoothing` argument equal to `FALSE`. This is equivalent to the previous procedure for survival and cumulative hazard estimation, but not for hazard and density function estimation. In the latter case, a smoothing bandwidth is also selected by `presmooth`, instead of being fixed by the user.

In boundary regions, hazard and density estimates corrected for boundary effects can be obtained by selecting one of "left", "right", or "both" options of the `bound` argument (see Mueller and Wang, 1994). Note that negative values of the estimates, a known problem with boundary kernels, are set to 0. For correcting the right-boundary effect, the maximum observed time is taken as the right endpoint of the support. Right-boundary correction should be used cautiously, due to the combined effect of the increased variance of the estimates and the small size of the risk set in the neighbouring of that end. With the default value of the `x.est` argument, estimation is restricted to values not greater than the 90th percentile of the observed times.

Value

An object of class 'survPresmooth'. Formally, it is a list with the following components:

`call`	The call the function was called with.
`data`	The data used, returned as a data frame if the value of `control$save.data` is `TRUE`.
`q.weight`	A numeric vector of length 2 giving the quantiles chosen as the ends of the support of the weight function.

bw.selec	The value of the bw.selec argument.
mise	A vector or matrix with the values of the bootstrap MISE, returned for bootstrap bandwidth selection if the value of control$save.mise is TRUE.
grid.pil	The vector of numeric values defining the grid used for searching the pilot bandwidth when plug-in or bootstrap bandwidth selection is done.
pilot.bw	The pilot bandwidth(s) used when plug-in or bootstrap bandwidth selection is done.
bandwidth	The bandwidth(s) selected.
grid.bw	A list of length 1 or 2 whose elements define the grid used for searching the bootstrap bandwidth.
p.hat	Nadaraya-Watson estimate of the conditional probability of non-censoring at the observed times.
x.est	A numeric vector with the points where estimates have been computed.
estimand	The input for the estimand argument.
estimate	A numeric vector with the presmoothed estimates at points x.est.

Author(s)

Ignacio Lopez-de-Ullibarri [aut, cre], Maria Amalia Jacome [aut]

References

Cao, R. and Jacome, M. A. (2004) "Presmoothed kernel density estimation for censored data", Journal of Nonparametric Statistics, 16, 289-309. doi: 10.1080/10485250310001622622.

Cao, R., Lopez-de-Ullibarri, I., Janssen, P. and Veraverbeke, N. (2005) "Presmoothed Kaplan-Meier and Nelson-Aalen estimators", Journal of Nonparametric Statistics, 17, 31-56. doi: 10.1080/10485250410001713981.

Cao, R. and Lopez-de-Ullibarri, I. (2007) "Product-type and presmoothed hazard rate estimators with censored data", Test, 16, 355-382. doi: 10.1007/s117490060014x.

Gonzalez-Manteiga, W., Cao R. and Marron J. S. (1996) "Bootstrap selection of the smoothing parameter in nonparametric hazard rate estimation", Journal of the American Statistical Association, 91, 1130-1140. doi: 10.1080/01621459.1996.10476983.

Jacome, M. A., Gijbels, I. and Cao, R. (2008) "Comparison of presmoothing methods in kernel density estimation under censoring", Computational Statistics, 23, 381-406. doi: 10.1007/s00180-00700766.

Lopez-de-Ullibarri, I and Jacome, M. A. (2013). "survPresmooth: An R Package for Presmoothed Estimation in Survival Analysis", Journal of Statistical Software, 54(11), 1-26. doi: 10.18637/jss.v054.i11.

Mueller, H. G. and Wang, J. L. (1994) "Hazard rate estimation under random censoring with varying kernels and bandwidths", Biometrics, 50, 61-76. doi: 10.2307/2533197.

See Also

control.presmooth

Examples

```
## Not run:
## Analysis with the example dataset (pscheck)

###############################################
## Cumulative hazard function (chf) estimation
###############################################

## Presmoothed estimate of chf with bootstrap bandwidth (fixing the
## randomization seed makes further comparisons easier)

set.seed(1)

Hboot1 <- presmooth(t, delta, pscheck, estimand = "H", bw.selec =
"bootstrap")

## As above, but: 1) specifying the points where the estimate is computed
## (note the warning), 2) specifying the search grid for the bandwidth,
## and 3) saving the bootstrap MISE

set.seed(1)

Hboot2 <- presmooth(t, delta, pscheck, estimand = "H", bw.selec =
"bootstrap", x.est = seq(0, 1, by = 0.02), grid.bw = seq(0.55, 0.7, by =
0.01), control = control.presmooth(save.mise = TRUE))

## A plot of the MISE, showing the bootstrap bandwidth

with(Hboot2,{
  plot(grid.bw$grid.bw, mise, xlab = "Bandwidth", ylab = "MISE", main =
  expression(paste("Bootstrap bandwidth, ", b[boot])), type = "l")
  points(bandwidth, mise[grid.bw$grid.bw == bandwidth], pch = 46, cex = 5)
  segments(bandwidth, 0, bandwidth, mise[grid.bw$grid.bw == bandwidth],
  lty = "dotted")
  text(bandwidth, min(mise), bquote(paste("  ", b[boot] ==
  .(bandwidth))), adj = c(0, 0))
})

## A plot of the presmoothed chf compared with Nelson-Aalen estimate and
## the true curve. The point (0, 0) must be added.

plot(c(0, Hboot2$x.est), c(0, Hboot2$estimate), xlab = "t", ylab =
"Cumulative hazard", type = "s")

Hna <- presmooth(t, delta, pscheck, estimand = "H", bw.selec = "fixed",
fixed.bw = 0)

lines(c(0, Hna$x.est), c(0, Hna$estimate), type = "s", col = "red")

curve(x^3, add = TRUE, col = "grey", lty = "dotted")

legend("topleft", legend = c("Presmoothed Nelson-Aalen", "Nelson-Aalen",
```

```
"True"), col = c("black", "red", "grey"), lty = c("solid", "solid",
"dotted"))

## An alternative way of obtaining the Nelson-Aalen estimate

Hna <- presmooth(t, delta, pscheck, "H", presmoothing = FALSE)

#################################
## Hazard function (hf) estimation
#################################

## (An example where right-boundary correction is successful)
## Presmoothed estimate of hf:
## 1) with plug-in bandwidth with and without right-boundary correction,
## 2) specifying the grid for presmoothing bandwidth selection, and
## 3) specifying the support of the weight function

hpi1 <- presmooth(t, delta, pscheck, estimand = "h", bw.selec =
"plug-in", x.est =  seq(0, max(pscheck$t), by = 0.02), grid.bw.pil =
seq(range(pscheck$t)[1],
range(pscheck)[2], by = 0.01), control = control.presmooth(q.weight =
c(0.25, 0.75)))

hpi2 <- presmooth(t, delta, pscheck, estimand = "h", bw.selec =
"plug-in", bound = "right", x.est = seq(0, max(pscheck$t), by = 0.02),
grid.bw.pil = seq(range(pscheck$t)[1], range(pscheck$t)[2], by = 0.01),
control = control.presmooth(q.weight = c(0.25, 0.75)))

plot(hpi1$x.est, hpi1$estimate, xlab = "t", ylab = "Hazard", ylim = c(0,
max(pmax(hpi1$estimate, hpi2$estimate))), type = "l")

lines(hpi2$x.est, hpi2$estimate, col = 2)

legend("bottomright", legend = c("none", "right"), title =
"Boundary effect correction", col = 1:2, lty = 1)

## Estimation of hf using a bootstrap bandwidth. The values chosen for
## the grid.bw argument are based on the result of preliminary trials
## (Warning: it may take a while ...)

set.seed(1)

hboot <- presmooth(t, delta, pscheck, estimand = "h", bw.selec =
"bootstrap", bound = "right", x.est = seq(0, max(pscheck$t), by = 0.02),
grid.bw.pil = seq(range(pscheck$t)[1],
range(pscheck)[2], by = 0.01), grid.bw = list(seq(0.4, 0.8, by = 0.05),
seq(0.6, 0.9, by = 0.005)), control = control.presmooth(q.weight =
c(0.25, 0.75), save.mise = TRUE))

## A plot of the bootstrap MISE, showing the bootstrap bandwidth

with(hboot, {
  contour(grid.bw$grid.bw.1, grid.bw$grid.bw.2, mise, levels
```

```
  = seq(min(mise), max(mise), length.out = 20), xlab =
  "Presmoothing bandwidth", ylab = "Smoothing bandwidth", main =
  "Bootstrap MISE")
  points(bandwidth[1], bandwidth[2], pch = 46, cex = 6)
  text(bandwidth[1], bandwidth[2], bquote(paste("  ", b[boot],
  symbol("="), symbol("("), .(bandwidth[1]), symbol(","), .(bandwidth[2]),
  symbol(")"))), adj = c(1, 0))
}
)

## Compare with the hf estimate obtained with plug-in bandwidth and the
## true curve

plot(hpi2$x.est, hpi2$estimate, xlab = "t", ylab = "Hazard", ylim = c(0,
max(pmax(hpi2$estimate, hboot$estimate))), type = "l")

lines(hboot$x.est, hboot$estimate, col = 2)

curve(3*x^2, add = TRUE, col = "grey", lty = "dotted")

legend("bottomright", legend = c("Plug-in bandwidth",
"Bootstrap bandwidth", "True"), col = c("black", "red", "grey"), lty =
c("solid", "solid", "dotted"))

###################################
## Density function (df) estimation
###################################

## Presmoothed estimate of df with plug-in and bootstrap bandwidths
## (with default options) and comparison with the true df

dpi <- presmooth(t, delta, pscheck, estimand = "f", bw.selec = "plug-in")

## The bootstrap presmoothing bandwidth is on the right boundary of the
## default grid (which in fact is the upper bound for the bandwidth: the
## range of the observed times)

set.seed(1)

dboot <- presmooth(t, delta, pscheck, estimand = "f", bw.selec =
"bootstrap")

## For this example, the estimates with either plugin or bootstrap
## bandwidth are very similar

plot(dpi$x.est, dpi$estimate, xlab = "t", ylab = "Density", ylim = c(0,
max(pmax(dpi$estimate, dboot$estimate))), type = "l", col = "blue",
lty = 2)

lines(dboot$x.est, dboot$estimate, col = "red", lty = 4)

curve(3*x^2*exp(-x^3), add = TRUE, lty = 1)
```

```
legend("bottomright", legend = c("Plug-in bandwidth",
"Bootstrap bandwidth", "True"), col = c("blue", "red", "black"), lty =
c(2, 4, 1))

## End(Not run)
```

`print.survPresmooth` *Print a survPresmooth Object*

Description

Print method for 'survPresmooth' objects.

Usage

```
## S3 method for class 'survPresmooth'
print(x, long = FALSE, more = NULL,
...)
```

Arguments

`x`	An object of class 'survPresmooth'.
`long`	A logical value. If `TRUE` the entire estimated curve is printed; if `FALSE`, the default, only the head of the estimated curve is printed if it consists of more than 100 values.
`more`	A character vector specifying the names of additional components of the `x` object that will be printed. The valid values for the components of the character vector are "data", "q.weight", "mise", "grid.pil", "pilot.bw", "grid.bw" and "p.hat". The default is `NULL`.
`...`	Optional arguments for the default method (i.e., `print.default`) of the `print` generic function.

Author(s)

Ignacio Lopez-de-Ullibarri [aut, cre], Maria Amalia Jacome [aut]

See Also

`presmooth`

Examples

```
## Not run:
## Printing an object of class 'survPresmooth' with 4 digits, including
## in the output the value of the pilot bandwidth and the estimated p
## function

print(presmooth(t, delta, pscheck), more = c("pilot.bw", "p.hat"),
```

```
digits = 4)

## End(Not run)
```

---

pscheck *Example Dataset*

Description

An artificial data frame used to illustrate the techniques implemented in the package. Pseudorandom numbers were drawn from Weibull distributions of scale parameter 1 and shape parameter 3 (for failure times) and 5 (for censoring times). Each observed time is the minimum of the corresponding failure and censoring times, and its censoring status indicates a failure time smaller than a censoring time (see `format` below).

Usage

```
pscheck
```

Format

The data frame contains 2 variables:

`t` Observed time.

`delta` Censoring status (0 = censored, 1 = uncensored).

Author(s)

Ignacio Lopez-de-Ullibarri [aut, cre], Maria Amalia Jacome [aut]

Index