Package: dpasurv (via r-universe)

November 2, 2024

Type Package

Title Dynamic Path Analysis of Survival Data via Aalen's Additive
Hazards Model

Version 0.1.0

Maintainer Matthias Kormaksson <mk375@cornell.edu>

Author Novartis Pharma AG [cph], Matthias Kormaksson [aut, cre],
Susanne Strohmaier [aut], Markus Lange [aut], David Demanse
[aut]

Description Dynamic path analysis with estimation of the corresponding
direct, indirect, and total effects, based on Fosen et al.,
(2006) <doi:10.1007/s10985-006-9004-2>. The main outcome of
interest is a counting process from survival analysis (or
recurrent events) data. At each time of event, ordinary linear
regression is used to estimate the relation between the
covariates, while Aalen's additive hazard model is used for the
regression of the counting process on the covariates.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.6.0)

Imports dplyr (>= 0.8.3), tidyr (>= 0.8.3), rlang (>= 0.4.0), ggplot2
(>= 3.2.0), survival (>= 2.44.1.1), timereg (>= 1.9.4), methods

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL http://opensource.nibr.com/dpasurv/

NeedsCompilation no

Repository CRAN

Date/Publication 2024-09-02 12:00:05 UTC

Contents

---

add.ci	*Calculate bootstrap confidence bands for effect of interest*

Description

Calculate bootstrap confidence bands for effect of interest

Usage

```
add.ci(object, alpha)
```

Arguments

object	object of class "effect"
alpha	the confidence level

Value

object of class "effect" with updated confidence interval corresponding to alpha

Examples

```
library(dpasurv)

data(simdata)

# Perform dynamic path analysis:
# We set boot.n=30 for the example to run fast, should be set large enough
# so that results don't change meaningfully for different seeds.
s <- dpa(Surv(start,stop,event)~M+x, list(M~x), id="subject", data=simdata, boot.n=30)

# Calculate cumulative direct effect (which calculates a CI with alpha = 0.05 by default):
direct <- effect(x ~ outcome, s)

# update confidence interval for a new alpha (this overwrites the 0.05 CI already calculated above)
direct <- add.ci(direct, alpha=0.10)
```

---

dpa	*Dynamic Path Analysis*

Description

Dynamic Path Analysis

Usage

```
dpa(
  out.formula,
  mediator.formulas,
  id,
  data,
  boot.n = 200,
  method = "timereg",
  progress_bar = FALSE,
  ...
)
```

Arguments

`out.formula`	Survival formula for Aalen's additive hazards model.
`mediator.formulas`	
	Mediator regression formula (in case of a single mediator), or a list of regression formulas (in case of multiple mediators). The formulas must be ordered according to Directed Acyclic Graph Structure (see Details).
`id`	character string indicating which column of 'data' corresponds to the subject ID. Bootstrapping will be performed on this id.
`data`	Data set in counting process format. In particular the data should contain a "start", "stop" and "event" column along with any mediators and baseline covariates.
`boot.n`	Number of bootstrap samples.
`method`	The underlying implementation of Aalen's additive regression model. Defaults to "timereg", which relies on the timereg::aalen() implementation, while method = "aareg" uses the survival::aareg() implementation.
`progress_bar`	Boolean. If TRUE, show progress bar. Defaults to FALSE.
`...`	other parameters passed to the Aalen's additive hazards model implementation. If method = "timereg", then ... will be passed to timereg::aalen(), while if method = "aareg", then ... will be passed to survival::aareg(). If ... contains parameters that don't belong to the formalArgs of the corresponding implementation then those parameters will be ignored.

Details

dpa performs Dynamic Path Analysis of a Directed Acyclic Graph (DAG). The out.formula can have as covariates all mediators listed in mediator.formulas. The mediator.formulas must obey the following DAG structure rule: The response of the k-th formula cannot appear as covariate in any of the formulas k+1, ..., length(mediator.formulas).

Value

Object of class 'dpa' with following fields:

coefs list of estimated coefficients from each of the regressions listed in out.formula and mediator.formulas.

boot.coefs list of bootstrap estimates corresponding to coefs. This stores all the bootstrap estimates to facilitate calculation of direct, indirect and total effects along with bootstrap confidence intervals.

meta a list keeping track of responses and covariates of each of the out.formula and mediator.formulas. Also keeps track of all variable types and level names in case of factors.

aalen Object storing information pertaining to the Aalen's additive model fit. Object is of class "aalen" if method="timereg", and of class "aareg" if method="aareg".

Examples

```
library(dpasurv)

data(simdata)

set.seed(1)

# Perform dynamic path analysis:
# We set boot.n=30 for the example to run fast, should be set large enough
# so that results don't change meaningfully for different seeds.
s <- dpa(Surv(start,stop,event)~M+x, list(M~x), id="subject", data=simdata, boot.n=30)

# Calculate cumulative direct, indirect, and total effects:
direct <- effect(x ~ outcome, s)
indirect <- effect(x ~ M ~ outcome, s)
total <- sum(direct, indirect)

# Plot the effects using basic graphics:
layout1x3 <- par(mfrow=c(1,3))
plot(direct); abline(h=0, lty=2, col=2)
plot(indirect); abline(h=0, lty=2, col=2)
plot(total); abline(h=0, lty=2, col=2)

# restore user's graphical parameters:
par(layout1x3)

# Plot the effects using ggplot2 graphics:
ggplot.effect(list(direct, indirect, total))
```

---

effect   *Effect estimation in dynamic path analysis*

Description

effect estimation method for class "dpa"

Usage

```
effect(formula, object, alpha = 0.05)
```

Arguments

formula   the formula for the direct or indirect effect to be estimated. Should be of the form: covariate ~ outcome for direct effect of covariate on outcome, while it should be of the form: covariate ~ mediator ~ outcome for indirect effect of covariate on outcome mediated through mediator. Note that the word "outcome" is reserved for the survival outcome process, but the word "covariate" and "mediator" should match a corresponding variable name in the data input. Alternatively the form can be: covariate ~ mediator for direct effects of covariate on mediator, or: covariate ~ mediator1 ~ mediator2 for indirect effects of covariate on mediator2 mediated through mediator1.

object   object of class "dpa" (as obtained by calling the function dpa) from which the effect is to be estimated.

alpha   The confidence level of the bootstrap intervals

Value

object of class "effect" with following fields:

coefs data.frame containing the unique event times along with the calculated effect coefficients. For effects corresponding to a continuous variable this results in a single effect column. For factors with n.levels categories the data.frame contains n.levels-1 effect columns each representing the effect coefficient of a particular factor level (as compared to reference level).

lower data.frame of same dimension as coefs containing the lower confidence bands of the effects stored in coefs

upper data.frame of same dimension as coefs containing the upper confidence bands of the effects stored in coefs

boot.coefs data.frame with three columns: one column of bootstrap sample ID, a second column of unique event times (per bootstrap sample), and a third column of the estimated effect coefficients (per bootstrap sample). The storing of the effects per bootstrap sample facilitates calculation of bootstrap confidence intervals for sums of indirect and direct effects.

label effect label with path specification: "direct" for direct effect and "indirect" for indirect effect mediated through a path of mediator(s)

scale scale of effect coefficients in coefs, lower, upper: "cumulative" (for effects on outcome) or "identity" (for effects on mediators)

alpha confidence level of the bootstrap intervals

Examples

```
library(dpasurv)

data(simdata)

set.seed(1)

# Perform dynamic path analysis:
# We set boot.n=30 for the example to run fast, should be set large enough
# so that results don't change meaningfully for different seeds.
s <- dpa(Surv(start,stop,event)~M+x, list(M~x), id="subject", data=simdata, boot.n=30)

direct <- effect(x ~ outcome, s)
indirect <- effect(x ~ M ~ outcome, s)
total <- sum(direct, indirect)
```

---

ggplot.effect	*Plot effects from dynamic path analysis along with bootstrap confidence bands*

Description

plotting method for class "effect"

Usage

```
ggplot.effect(
  object,
  relative = FALSE,
  titles = NULL,
  x_label = "Time",
  y_label = NULL
)
```

Arguments

object	object of class "effect", or list of objects of class "effect"
relative	should the effect be plotted on a relative survival scale (i.e. 'y=exp(-effect)')?. Defaults to FALSE.
titles	If NULL, function will automatically generate. Otherwise character vector of length equal to number of elements in object list
x_label	Label for x-axis. Defaults to "Time"
y_label	Label for y-axis. Default when object scale is "cumulative" will be "Cumulative Effect" (relative=FALSE) and "Relative survival" (relative=TRUE). If object scale is "identity" then the default y_label will be "Effect".

Value

ggplot object

Examples

```
library(dpasurv)

data(simdata)

set.seed(1)

# Perform dynamic path analysis:
# We set boot.n=30 for the example to run fast, should be set large enough
# so that results don't change meaningfully for different seeds.
s <- dpa(Surv(start,stop,event)~M+x, list(M~x), id="subject", data=simdata, boot.n=30)

direct <- effect(x ~ outcome, s)
indirect <- effect(x ~ M ~ outcome, s)
total <- sum(direct, indirect)

ggplot.effect(direct)
ggplot.effect(list(direct, indirect, total))
```

---

plot.effect	*Plot effects from dynamic path analysis along with bootstrap confidence bands*

Description

plotting method for class "effect"

Usage

```
## S3 method for class 'effect'
plot(x, relative = FALSE, ...)
```

Arguments

x	object of class "effect"
relative	should the effect be plotted on a relative survival scale (i.e. 'y=exp(-effect)')?. Defaults to FALSE.
...	other graphical parameters passed to the graphics::plot function.

Value

this function does not return anything, but simply plots the associated effect encoded in the object x

Examples

```
library(dpasurv)

data(simdata)

set.seed(1)

# Perform dynamic path analysis:
# We set boot.n=30 for the example to run fast, should be set large enough
# so that results don't change meaningfully for different seeds.
s <- dpa(Surv(start,stop,event)~M+x, list(M~x), id="subject", data=simdata, boot.n=30)

direct <- effect(x ~ outcome, s)
indirect <- effect(x ~ M ~ outcome, s)
total <- sum(direct, indirect)

par(mfrow=c(1,3))
layout1x3 <- par(mfrow=c(1,3))
plot(direct); abline(h=0, lty=2, col=2)
plot(indirect); abline(h=0, lty=2, col=2)
plot(total); abline(h=0, lty=2, col=2)

# restore user's graphical parameters:
par(layout1x3)
```

---

simdata   *Simulated data*

Description

This is a simulated data set in the (start,stop] format. The subject column refers to subject ID and note that there are multiple rows per subject. The columns x and dose correspond to a treatment and dose variable respectively, while M refers to the longitudinal mediator values. The triplet (start, stop, event) corresponds to the time-to-event data in the required (start,stop] format.

Format

A data frame with the following variables: subject (fct), x (dbl), dose (fct), M (dbl), start (dbl), stop (dbl), event (dbl).

---

sum.effect   *Sum of direct and indirect effects from dynamic path analysis*

Description

a sum method for class "effect"

Usage

```
## S3 method for class 'effect'
sum(effect1, effect2, ...)
```

Arguments

effect1	an object of class "effect" obtained from a call to the function effect()
effect2	a second object of class "effect" obtained from a call to the function effect()
...	additional objects of class "effect" (any number of effects allowed)

Value

an object of class "effect" containing the sum of effect1, effect2, ...

Examples

```
library(dpasurv)

data(simdata)

set.seed(1)

# Perform dynamic path analysis:
# We set boot.n=30 for the example to run fast, should be set large enough
# so that results don't change meaningfully for different seeds.
s <- dpa(Surv(start,stop,event)~M+x, list(M~x), id="subject", data=simdata, boot.n=30)

direct <- effect(x ~ outcome, s)
indirect <- effect(x ~ M ~ outcome, s)
total <- sum(direct, indirect)
```

Index