Package ‘spsurv’

March 31, 2020

Type Package
Title Bernstein Polynomial Based Semiparametric Survival Analysis
Version 1.0.0
Maintainer Renato Panaro <renatovp@ime.usp.br>
License GPL-3
Biarch true
Depends R (>= 3.5.0), survival (>= 2.44-1.1), loo (>= 2.1.0), coda (>= 0.19-3), MASS (>= 7.3-51.4)
Imports methods, Rcpp (>= 0.12.0), rstan (>= 2.18.1), rstantools (>= 1.5.1)
LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)
Suggests knitr, devtools, roxygen2, testthat, KMsurv
Encoding UTF-8
LazyData true
SystemRequirements GNU make
BugReports https://github.com/rvpanaro/spsurv/issues
VignetteBuilder knitr
RoxygenNote 7.0.2
NeedsCompilation yes
Author Renato Panaro [aut, cre] (<https://orcid.org/0000-0002-1903-2091>, https://rvpanaro.github.io/), Fábio Demarqui [aut] (<https://orcid.org/0000-0001-9236-1986>), Vinícius Mayrink [aut] (<https://orcid.org/0000-0002-5683-8326>)
Repository CRAN
Date/Publication 2020-03-31 10:00:02 UTC
The 'spsurv' package.

Description

A set of flexible routines to allow semiparametric survival regression modeling based on Bernstein polynomial, including Bernstein based proportional hazards model (BPPH), Bernstein polynomial based proportional odds model (BPPO) and Bernstein based accelerated failure time model (BPAFT) for right-censored data.
Details

*bp.basis* fits semi-parametric models for time-to-event survival data. Non-informative right-censoring assumption is available. Any user-defined Bernstein polynomial can be user-defined using an arbitrary degree, i.e. highest basis polynomials order.

The framework takes advantage of fully likelihood methods since the polynomial parameters are used to estimate the baseline functions. Even so, this is said to be semi-parametric since this approach does not rely on any distribution. Unlike the Cox model, the BP based models provide smooth hazard and survival curve estimates.

Value

none

Author(s)

Renato Valladares Panaro (renatovp@ime.usp.br).

References


---

**bp.basis**

*Bernstein basis polynomials calculations*

**Description**

Bernstein basis polynomials calculations

**Usage**

bp.basis(time, degree, tau = max(time))

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>a vector of times.</td>
</tr>
<tr>
<td>degree</td>
<td>Bernstein polynomial degree</td>
</tr>
<tr>
<td>tau</td>
<td>must be greater than times maximum value observed.</td>
</tr>
</tbody>
</table>
Value
A list containing matrices b and B corresponding BP basis and corresponding tau value used to compute them.

Description
Fits the BPAFT model to time-to-event data.

Usage
bpaft(formula, degree, data, approach = c("mle", "bayes"), ...)

Arguments
formula a Surv object with time to event observations, right censoring status and explanatory terms.
degree Bernstein polynomial degree.
data a data.frame object.
approach Bayesian or maximum likelihood estimation methods, default is approach = "mle".
... further arguments passed to or from other methods

Value
An object of class 'spbp'.

See Also
spbp, bpph and bppo for other BP based models.

Examples

library("spsurv")
data("veteran")

fit <- bpaft(Surv(time, status) ~ karno + celltype, data = veteran)

summary(fit)
**bpph**

*Bernstein Polynomial Based Proportional Hazards Model*

**Description**

Fits the BPPH model to time-to-event data.

**Usage**

```r
bpph(formula, degree, data, approach = c("mle", "bayes"), ...)
```

**Arguments**

- `formula`: a Surv object with time to event observations, right censoring status and explanatory terms.
- `degree`: Bernstein polynomial degree.
- `data`: a data.frame object.
- `approach`: Bayesian or maximum likelihood estimation methods, default is approach = "mle".
- `...`: further arguments passed to or from other methods

**Value**

An object of class `spbp`.

**See Also**

`spbp`, `bppo` and `bpaft` for other BP based models.

**Examples**

```r
library("spsurv")
data("veteran")

fit <- bpph(Surv(time, status) ~ karno + factor(celltype), data = veteran)

summary(fit)
```
**bppo**

*Bernstein Polynomial Based Proportional Odds Model*

**Description**

Fits the BPPO model to time-to-event data.

**Usage**

```r
bppo(formula, degree, data, approach = c("mle", "bayes"), ...)
```

**Arguments**

- `formula` a `Surv` object with time-to-event observations, right censoring status and explanatory terms.
- `degree` Bernstein polynomial degree.
- `data` a `data.frame` object.
- `approach` Bayesian or maximum likelihood estimation methods, default is `approach = "mle"`.
- `...` further arguments passed to or from other methods

**Value**

An object of class `spbp`.

**See Also**

`sbsp`, `bpph` and `bpaft` for other BP based models.

**Examples**

```r
library("spsurv")
data("veteran")

fit <- bppo(Surv(time, status) ~ karno + celltype,
data = veteran)

summary(fit)
```
coef

Generic S3 method coef

Description
Generic S3 method coef

Usage
coef(spbp, ...)

Arguments

- spbp: a fitted model object
- \ldots: further arguments passed to or from other methods.

Value
the estimated regression coefficients

coef.spbp

Estimated regression coefficients

Description
Estimated regression coefficients

Usage
## S3 method for class 'spbp'
coef(spbp, ...)

Arguments

- spbp: an object of the class spbp
- \ldots: further arguments passed to or from other methods

Value
the estimated regression coefficients
confint

Generic S3 method confint

Usage

confint(spbp, ...)

Arguments

spbp a fitted model object
...

Value

the estimated regression coefficients

confint.spbp

Confidence intervals for the regression coefficients

Description

Confidence intervals for the regression coefficients

Usage

## S3 method for class 'spbp'
confint(spbp, level = 0.95, ...)

Arguments

spbp an object of the class spbp
level the confidence level required
...

Value

100(1-alpha) confidence intervals for the regression coefficients
extract  

Generic S3 method extract

Description
Generic S3 method extract

Usage
extract(spbp, ...)

Arguments
spbp: a fitted model object
...
... further arguments passed to or from other methods.

Value
extract the MCMC chain values of a Bayesian fit.

extract.spbp  

Extract method for fitted spbp models

Description
Extract samples from a fitted spbp model.

Usage
## S3 method for class 'spbp'
extract(spbp, pars = c("beta", "gamma"), ...)

Arguments
spbp: an object of class 'spbp' result of a spbp fit.
pars: parameters to be selected.
...
... arguments inherent from extract.

Value
see extract.

See Also
spbp, stan_dens.spbp, traceplot.spbp
Examples

library("spsurv")
data("veteran")

fit <- bpph(Surv(time, status) ~ karno + factor(celltype),
data = veteran)
extract(fit)

itsamp

Inverse Transform Sampling To Generate Time-to-event Data From Parametric Models

Description

Random survival times generation for the weibull or log-logistic distributions with parameters 'scale' and 'shape'.

Usage

itsamp(
  n,
  beta = c(2, -1),
  event_scale = 10,
  censor_scale = 4,
  features = data.frame(x1 = rnorm(n, 0), x2 = rnorm(n, 0)),
  shape = 2,
  model = c("ph", "po", "aft"),
  dist = c("weibull", "llogis"),
  censor = TRUE
)

Arguments

n integer; sample size
beta vector of regression coefficients
event_scale, censor_scale event and censoring scale parameters
features matrix of features (columns)
shape event and censoring distribution shape
model either "ph" (default) or "aft" for weibull and "po" or "aft" for log-logistic distribution
dist "weibull" or "llogis"
censor logical; if 'TRUE', censoring is required, that is mean(status) > 0
Details

sim_surv returns weibull (log-logistic) randomly generated survival times. According to Collett (2003), the accelerated failure time model encompasses a wide variety of parametric models, including weibull and log-logistic models.

Value

data.frame of `ncol(x) +2` columns in which the survival times are the response variable denoted by `y`, `status` indicates failure (0 = failure) and the features are appended to the next columns.

See Also

spbp

Examples

```r
rows <- 200
categorical <- rbinom(rows, size = 3, prob = .5)
x <- data.frame(numerical = rnorm(rows),
cat0 = as.numeric(categorical == 0),
cat1 = as.numeric(categorical == 1),
cat2 = as.numeric(categorical == 2),
cat3 = as.numeric(categorical == 3))

newdata <- itsamp(n = rows, beta = c(1, -2, .5, .1, 1),
features = x, model = 'ph', dist = 'weibull')
```

--

mode  

**Calculate the posterior mode**

Description

Calculate the posterior mode

Usage

mode(ext)

Arguments

ext  

rstan extracted sample.

Value

A vector containing the posterior mode of each sample.
model.matrix.spbp  Model.matrix method for fitted spbp models

Description

Model.matrix of a fitted spbp model.

Usage

```r
## S3 method for class 'spbp'
model.matrix(
  object,
  data = eval(object$call$data, envir = parent.frame()),
  ...
)
```

Arguments

- `object` an object of class ‘spbp’, see spbp.
- `data` data.frame object.
- `...` arguments inherent from model.matrix.

Value

The explanatory variables matrix.

See Also

spbp, model.matrix

Examples

```r
library("spsurv")
data("veteran")

fit <- bpph(Surv(time, status) ~ karno + factor(celltype),
            data = veteran)

model.matrix(fit)
```
Description

Bernstein Polynomial Based Regression Object Print

Usage

```r
## S3 method for class 'spbp'
print(
  x,
  digits = max(getOption("digits") - 4, 3),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

- `x` : an object of class `spbp`
- `digits` : number of digits to display
- `signif.stars` : see `getOption`
- `...` : further arguments passed to or from other methods

Value

none

Description

Bernstein Polynomial Based Regression Object Summary BPAFT Bayes

Usage

```r
## S3 method for class 'summary.bpaft.bayes'
print(...)
```

Arguments

- `...` : further arguments passed to or from other methods
print.summary.bpaft.mle

Bernstein Polynomial Based Regression Object Summary BPAFT MLE

Description
Bernstein Polynomial Based Regression Object Summary BPAFT MLE

Usage
## S3 method for class 'summary.bpaft.mle'
print(...)  

Arguments
...
  further arguments passed to or from other methods

Value
none

print.summary.bpph.bayes

Bernstein Polynomial Based Regression Object Summary BPPH Bayes

Description
Bernstein Polynomial Based Regression Object Summary BPPH Bayes

Usage
## S3 method for class 'summary.bpph.bayes'
print(...)  

Arguments
...
  further arguments passed to or from other methods

Value
none
print.summary.bpph.mle

Bernstein Polynomial Based Regression Object Summary BPPH MLE

Description
Bernstein Polynomial Based Regression Object Summary BPPH MLE

Usage
## S3 method for class 'summary.bpph.mle'
print(...)

Arguments
... further arguments passed to or from other methods

Value
none

print.summary.bppo.bayes

Bernstein Polynomial Based Regression Object Summary BPPO Bayes

Description
Bernstein Polynomial Based Regression Object Summary BPPO Bayes

Usage
## S3 method for class 'summary.bppo.bayes'
print(...)

Arguments
... further arguments passed to or from other methods

Value
none
print.summary.bpbo.mle

Bernstein Polynomial Based Regression Object BPPO MLE

Description
Bernstein Polynomial Based Regression Object BPPO MLE

Usage
## S3 method for class 'summary.bpbo.mle'
print(...)  

Arguments
... further arguments passed to or from other methods

Value
none

print.summary.spbp.bayes

Bernstein Polynomial Based Regression Object Summary Bayes

Description
Bernstein Polynomial Based Regression Object Summary Bayes

Usage
## S3 method for class 'summary.spbp.bayes'
print(x, digits = max(getOption("digits") - 4, 3), ...)  

Arguments
x a summary.spbp.bayes object
digits number of digits to display.
... further arguments passed to or from other methods

Value
none
Bernstein Polynomial Based Regression Object Summary MLE

Description
Bernstein Polynomial Based Regression Object Summary MLE

Usage

```r
## S3 method for class 'summary.spbp.mle'
print(
  x,
  digits = max(getOption("digits") - 4, 3),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

- `x`: a summary.spbp.mle object
- `digits`: number of digits to display.
- `signif.stars`: see `getOption`
- `...`: further arguments passed to or from other methods

Value
none

residuals.spbp

BP based models residuals.

Description
Residuals for a fitted `spbp` model.

Usage

```r
## S3 method for class 'spbp'
residuals(object, type = c("cox-snell"), ...)
```

Arguments

- `object`: an object of class `spbp` result of a `spbp` fit.
- `type`: type of residuals, default is "cox-snell"
- `...`: further arguments passed to or from other methods
spbp

See Also

spbp.

Examples

library("spsurv")
data("veteran")

fit <- bpph(Surv(time, status) ~ karno + factor(celltype),
           data = veteran)
residuals(fit)

spbp

spbp: The BP Based Survival Analysis Function

Description

Semiparametric Survival Analysis Using Bernstein Polynomial

Usage

spbp(formula, ...)

Arguments

formula  a Surv object with time to event, status and explanatory terms.
...      Arguments passed to ‘rstan::sampling’ (e.g. iter, chains) or ‘rstan::optimizing’.

Details

Fits Bernstein Polynomial based Proportional regression to survival data.

Value

An object of class `spbp`.

See Also

spbp.default
spbp.default, bpph, bppo, bpaft, https://mc-stan.org/users/documentation/
Examples

library("spsurv")
data("veteran") ## imports from survival package

fit_mle <- spbp(Surv(time, status) ~ karno + factor(celltype),
data = veteran, model = "po")
summary(fit_mle)

fit_bayes <- spbp(Surv(time, status) ~ karno + factor(celltype),
data = veteran, model = "po", approach = "bayes",
cores = 1, iter = 300, chains = 1,
priors = list(beta = c("normal(0,4)"),
           gamma = "lognormal(0,4)"))

summary(fit_bayes)

spbp.default

Description

spbp: The BP Based Semiparametric Survival Analysis Function

Usage

## Default S3 method:
spbp(
  formula,
  degree,
  data,
  approach = c("mle", "bayes"),
  model = c("ph", "po", "aft"),
priors = list(beta = c("normal(0,4)"), gamma = "lognormal(0,10)"),
scale = TRUE,
cores = parallel::detectCores(),
...
)

Arguments

formula a Surv object with time to event, status and explanatory terms
degree Bernstein Polynomial degree
data a data.frame object
approach Bayesian or Maximum Likelihood estimation methods, default is approach = "bayes"
### stan_dens.spbp

**Description**

Posterior density of samples from a fitted spbp model.

**Usage**

```r
## S3 method for class 'spbp'
stan_dens(spbp, pars = c("beta", "gamma"), ...)
```

**Arguments**

- `spbp`: a fitted model object
- `pars`: further arguments passed to or from other methods.

**Value**

the density plot of a MCMC chain.
Arguments

- `spbp` the result of a `spbp` fit.
- `pars` parameters to be selected.
- `...` arguments inherent from `stan_dens`.

Value

see `stan_dens`.

See Also

`spbp`, `traceplot.spbp`, `extract.spbp`

Examples

```r
library("spsurv")
data("veteran")

fit <- bpph(Surv(time, status) ~ karno + factor(celltype),
data = veteran)

stan_dens(fit)
```

summary.spbp  

Bernstein Polynomial Based Regression Object Summary

Description

Bernstein Polynomial Based Regression Object Summary

Usage

```r
## S3 method for class 'spbp'
summary(object, interval = 0.95, ...)
```

Arguments

- `object` an object of class `spbp`
- `interval` interval coverage (confidence or credibility)
- `...` further arguments passed to or from other methods

Value

An object of class analogous to for e.g. `summary.bppo.bayes`. 
survivor

Description
Generic S3 method \texttt{vcov}

Usage
\texttt{survivor(spbp, \ldots)}

Arguments
\begin{itemize}
  \item \texttt{spb}b an object of class \texttt{spb}p
  \item \texttt{\ldots} further arguments passed to or from other methods
\end{itemize}

Value
estimates survival for each dataset individual (line). Spbp Object Observed Survival

See Also
\texttt{spb}, \texttt{itsamp}

traceplot

Description
Generic S3 method \texttt{traceplot}

Usage
\texttt{traceplot(spbp, \ldots)}

Arguments
\begin{itemize}
  \item \texttt{spb}p a fitted model object
  \item \texttt{\ldots} further arguments passed to or from other methods.
\end{itemize}

Value
the traceplot of a MCMC chain.
Description

Traceplot of a Bayesian fit \texttt{spbp}.

Usage

\begin{verbatim}
## S3 method for class 'spbp'
traceplot(spbp, pars = c("beta", "gamma"), ...)
\end{verbatim}

Arguments

- \texttt{spbpb} an object of class \texttt{spbp} result of a \texttt{spbp} fit.
- \texttt{pars} parameters to be selected.
- \texttt{...} arguments inherent from \texttt{traceplot}.

Value

see \texttt{traceplot}.

See Also

\texttt{spbp, stan_dens.spbp, extract.spbp}

Examples

\begin{verbatim}
library("spsurv")
data("veteran")

fit <- bpph(Surv(time, status) ~ karno + factor(celltype),
            data = veteran)

traceplot(fit)
\end{verbatim}
**vcov**

*Generic S3 method vcov*

**Description**
Generic S3 method vcov

**Usage**
vcov(spbp, ...)

**Arguments**

- **spbp**
  - a fitted model object
- **...**
  - further arguments passed to or from other methods.

**Value**
the variance-covariance matrix associated the regression coefficients.

---

**vcov.spbp**

*Covariance of the regression coefficients*

**Description**
Covariance of the regression coefficients

**Usage**

```r
## S3 method for class 'spbp'
vcov(spbp, ...)
```

**Arguments**

- **spbp**
  - an object of the class spbp
- **...**
  - further arguments passed to or from other methods.

**Value**
the variance-covariance matrix associated with the regression coefficients.
Index

bp.basis, 3
bpaft, 4, 5, 6, 18
bpph, 4, 5, 6, 18
bppo, 4, 5, 6, 18

coef, 7
coeff.spbp, 7
confint, 8
confint.spbp, 8

eextract, 9, 9
extract.spbp, 9, 21, 23

ggetOption, 13, 17

itsamp, 10, 22

mode, 11
model.matrix, 12
model.matrix.spbp, 12

print.spbp, 13
print.summary.bpaft.bayes, 13
print.summary.bpaft.mle, 14
print.summary.bpph.bayes, 14
print.summary.bpph.mle, 15
print.summary.bppo.bayes, 15
print.summary.bppo.mle, 16
print.summary.spbp.bayes, 16
print.summary.spbp.mle, 17

residuals.spbp, 17

spbp, 3–6, 9, 11, 12, 17, 18, 20–23
spbp.default, 18, 19
spsurv(spsurv-package), 2
spsurv-package, 2
stan_dens, 20, 21
stan_dens.spbp, 9, 20, 23
summary.spbp, 21
survivor, 22

traceplot, 22, 23
traceplot.spbp, 9, 21, 23

vcov, 24
vcov.spbp, 24