Package 'lpl'

July 22, 2025

Type Package

Title Local Partial Likelihood Estimation and Simultaneous Confidence Band

Version 0.12

Date 2025-04-01

Author Bingshu E. Chen [aut, cre],
Yicong Liu [aut],
Siwei Zhang [aut],
Teng Wen [aut],
Wenyu Jiang [aut]

Maintainer Bingshu E. Chen <bingshu.chen@queensu.ca>

Depends R (>= 3.5.0), MASS, methods, parallel, survival

Description Local partial likelihood estimation by Fan, Lin and Zhou(2006)<doi:10.1214/009053605000000796> and simultaneous confidence band is a set of tools to test the covariates-biomarker interaction for survival data. Test for the covariates-biomarker interaction using the bootstrap method and the asymptotic method with simultaneous confidence band (Liu, Jiang and Chen (2015)<doi:10.1002/sim.6563>).

License GPL-2

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2025-04-04 16:40:02 UTC

Contents

`lpl-package` *Local Partial Likelihood Boostrap test*

Description

This package fits a multivariable local partial likelihood model for covariate-biomarker interaction with survival data.

Details

"lpl" is a R package for multivariate covariate-biomarker interaction uisng local partial likelihood method.

Please use the following steps to install 'lpl' package:

1. First, you need to install the 'devtools' package. You can skip this step if you have 'devtools' installed in your R. Invoke R and then type

install.packages("devtools")

2. Load the devtools package.

library(devtools)

3. Install "lpl" package with R commond

install_github("statapps/lpl")

"lpl" uses local partial likelihood to etimate covariate-biomarker interactions and bootstrap method to test the significance of the interactions.

Author(s)

Siwei Zhang and Bingshu E. Chen

Maintainer: Bingshu E. Chen <bingshu.chen@queensu.ca>

References

1. Fan, J., Lin, H,, Zhou, Y. (2006). Local partial-likelihood estimation for lifetime data. The Annals of Statistics. 34, 290-325.

2. Liu, Y., Jiang, W. and Chen, B. E. (2015). Testing for treatment-biomarker interaction based on local partial-likelihood. Statistics in Medicine. 34, 3516-3530.

3. Zhang, S., Jiang, W. and Chen, B. E. (2016). Estimate and test of multivariate covariates and biomarker interactions for survival data based on local partial likelihood. Manuscript in preparation.

See Also

`coxph, survival`

Examples

```
# fit = lpl(y~trt+age+biomarker)
```

`control` *Auxiliary function for lpl fitting*

Description

Auxiliary function for `lple` fitting. Typically only used internally by 'lpl', but may be used to construct a control argument to either function.

Usage

```
#   lpl.control(h, kernel = 'gaussian', B, w0, p1, pctl)
```

Arguments

`h`	bandwidth of kernel function. The default value is h = 0.2
`kernel`	kernel funtion types, including "gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "cosine", "optcosine". The default value is 'gaussian'
`B`	number of bootstrap times. The default value is 200
`w0`	the estimated points in the interval of (0,1), select arbitrarily. The default value is seq(0.05, 0.95, 0.025)
`p1`	the number of dependend variables that make interactions with the biomarker w. The default value is 1
`pctl`	the estimated points that want to be shown in the output. The default value is seq(0.2, 0.8, 0.1)

Details

Control is used in model fitting of lpl.

Value

This function checks the internal consisitency and returns a list of value as inputed to control model fit of lpl.

Author(s)

Siwei Zhang and Bingshu E. Chen

See Also

`lplb`, `lple`

Examples

```
## The default control values are: h = 0.2, kernel = 'gaussian', B = 200,
## w0 = seq(0.05, 0.95, 0.025), p1 = 1, pctl = seq(0.2, 0.8, 0.1)
##
## To fit the lpl model with some control variables changed,

w0=seq(0.05, 0.95, by=0.05)
ctl = lpl.control(w0=w0, h=0.3, p1=2, B=100)

## then fit the lple model
```

`lplb` *Local partial likelihood bootstrap (LPLB) method to fit biomarker Models*

Description

{lplb} is a R package for local partial likelihood estimation (LPLE) (Fan et al., 2006) of the coefficients of covariates with interactions of the biomarker W, and hypothesis test of whether the relationships between covariates and W are significant, by using bootstrap method.

Usage

```
## Default S3 method:
lplb(x, y, control, ...)
## S3 method for class 'formula'
lplb(formula, data=list(...), control = list(...), ...)

# use
#          lplb(y ~ X1+X2+...+Xp+w, data=data, control)
#
# to fit a model with interactions between biomarker (w) with the first p1
# terms of dependent variables.
# p1 is included in 'control'. p1<p. See 'lplb.control' for details
#
# use
#         lplb(x, y, control)
#
# to fit a model without the formula
#
# Biomarker w should be the 'LAST' dependend variable
```

Arguments

`formula`	an object of class "formula"(or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.

data	an optional data frame, list or environment (or object coercible by 'as.data.frame' to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula).
x, y	For 'lplb.default', x is a design matrix of dimension n * (p+1) and y is a vector of observations of length n for a "Surv" object for "coxph".
control	a list of parameters for controlling the fitting process. See 'lplb.control' for details
...	additional arguments to be passed to the low level regression fitting functions (see below).

Details

Here 'w' is a Biomarker variable. This variable is required and shall be the last dependent variable in the formula.

'x.cdf' is a function that maps biomarker values to interval (0, 1) using its empirical cumulative distribution function.

Value

lplb returns an object of class inheriting from "lplb" which inherits from the class 'coxph'. See later in this section.

The function "print" (i.e., "print.lplb") can be used to obtain or print a summary of the results.

An object of class "lplb" is a list containing at least the following components:

beta_w	a matrix of m * p1, the estimated coefficients at each of the m estimated points, for the first p1 dependent variables with interactions of the biomarker w
Q1	the test statistic of the data
mTstar	a vector of the test statistics from B times' bootstrap
pvalue	the p-value of the hypothesis that beta_w is a constant

Note

This package was build on code developed by Yicong Liu for simple treatment-biomaker interaction model.

Author(s)

Siwei Zhang and Bingshu E. Chen (bingshu.chen@queensu.ca)

References

Zhang, S., Jiang, W. and Chen, B. E. (2016). Estimate and test of multivariate covariates and biomarker interactions for survival data based on local partial likelihood. Manuscript in preparation.

See Also

coxph, lpl.control print.lple plot.lple

Examples

```
dat = lplDemoData(50)
fit = lplb(Surv(time, status)~z1 + z2 + w, data = dat, B = 3, p1 = 2)
print(fit)
```

`lple` *Local partial likelihood estimate (LPLE) method to fit biomarker Models*

Description

{lple} is a R package for local partial likelihood estimation (LPLE) (Fan et al., 2006) of the coefficients of covariates with interactions of the biomarker W, and hypothesis test of whether the relationships between covariates and W are significant, by using bootstrap method.

Usage

```
## Default S3 method:
lple(x, y, control, ...)
## S3 method for class 'formula'
lple(formula, data=list(...), control = list(...), ...)

# use
#          lple(y ~ X1+X2+...+Xp+w, data=data, control)
#
# to fit a model with interactions between biomarker (w) with the first p1
# terms of dependent variables.
# p1 is included in 'control'. p1<p. See 'lplb.control' for details
#
# use
#         lple(x, y, control)
#
# to fit a model without the formula
#
# Biomarker w should be the 'LAST' dependend variable
```

Arguments

`formula`	an object of class "formula"(or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
`data`	an optional data frame, list or environment (or object coercible by 'as.data.frame' to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula).
`x, y`	For 'lple.default', x is a design matrix of dimension n * (p+1) and y is a vector of observations of length n for a "Surv" object for "coxph".

control	a list of parameters for controlling the fitting process. See 'lplb.control' for details
...	additional arguments to be passed to the low level regression fitting functions (see below).

Details

Here 'w' is a Biomarker variable. This variable is required and shall be the last dependent variable in the formula.

'x.cdf' is a function that maps biomarker values to interval (0, 1) using its empirical cumulative distribution function.

Value

lple returns an object of class inheriting from "lple" which inherits from the class 'coxph'. See later in this section.

The function "print" (i.e., "print.lple") can be used to obtain or print a summary of the results.

An object of class "lple" is a list containing at least the following components:

beta_w	a matrix of m * p1, the estimated coefficients at each of the m estimated points, for the first p1 dependent variables with interactions of the biomarker w
Q1	the test statistic of the data
mTstar	a vector of the test statistics from B times' bootstrap
pvalue	the p-value of the hypothesis that beta_w is a constant

Note

This package was build on code developed by Yicong Liu for simple treatment-biomaker interaction model.

Author(s)

Siwei Zhang and Bingshu E. Chen (bingshu.chen@queensu.ca)

References

Zhang, S., Jiang, W. and Chen, B. E. (2016). Estimate and test of multivariate covariates and biomarker interactions for survival data based on local partial likelihood. Manuscript in preparation.

See Also

coxph, lpl.control print.lple plot.lple

Examples

```
dat = lplDemoData(50)
fit = lple(Surv(time, status)~z1 + w, data = dat, p1 = 1)
print(fit)
predict(fit)
survfit(fit, se.fit = FALSE)
```

`plot.lple` *The Plot Function of lple*

Description

Draw a series of plots of beta_w vs. w_est for each dependent variable with interactions with the biomarker w. See also: `lple`, `lpl.control`

Usage

```
## S3 method for class 'lple'
plot(x, ..., scale = c('original', 'transformed'))
```

Arguments

`x`	a lple class returned from lple fit.
`scale`	choose the scale of biomarker variable, 'original' or 'o' for the original biomarker scale. 'transformed' or 't' for transformed scale that mapps biomarker to interval (0, 1). The default is to plot in the original scale.
`...`	other options used in plot().

Details

plot.lple is called to plot the relationships between beta_w and w_est for each dependent variable with interactions with the biomarker w, from the `lple` fit model.

The number of interaction terms can be set in `lpl.control`.

The default method, print.default has its own help page. Use methods("print") to get all the methods for the print generic.

Value

No return value, called for plot model fit

Author(s)

Bingshu E. Chen and Siwei Zhang

See Also

`lplb`, `lple`, `lpl.control`, `print.lple`

Examples

```
dat = lplDemoData(50)
fit = lple(Surv(time, status)~z1 + w, data = dat, p1 = 1)
plot(fit)
```

`predict.lple` *predict a lple object*

Description

Compute fitted values and prediction error for a model fitted by lple

Usage

```
## S3 method for class 'lple'
## S3 method for class 'lple'
predict(object, newdata, newy=NULL, ...)
## S3 method for class 'lple'
residuals(object, type=c("martingale", "deviance"), ...)
```

Arguments

`object`	a model object from the lple fit
`newdata`	optional new data at which to do predictions. If absent, predictions are for the dataframe used in the original fit
`newy`	optional new response data. Default is NULL
`type`	type of residuals, the default is a martingale residual
`...`	additional arguments affecting the predictions produced

Details

predict.lple is called to predict object from the lple model `lple`.

The default method, predict has its own help page. Use methods("predict") to get all the methods for the predict generic.

Value

predict.lple returns a list of predicted values, prediction error and residuals.

`lp`	linear predictor of beta(w)*Z, where beta(w) is the fitted regression coefficient and Z is covariance matrix.
`risk`	risk score, exp(lp). When new y is provided, both lp and risk will be ordered by survival time of the new y.
`residuals`	martingale residuals of the prediction, if available.

pe.mres	prediction error based on martingale residual, if both new data and new y is provided.
cumhaz	cumulative hzard function.
time	time for cumulative hazard function. Time from new y will be used is provided

Author(s)

Bingshu E. Chen

See Also

The default method for predict predict,

For the Cox model prediction: predict.coxph. #survfit.lple

---

print.lplb *print a lplb object*

Description

print are used to provide a short summary of lplb outputs.

Usage

```
## S3 method for class 'lplb'
print(x, ...)
```

Arguments

x	a lplb class returned from lplb fit
...	other options used in print()

Details

print.lplb is called to print object or summary of object from the lplb model lplb.

The default method, print.default has its own help page. Use methods("print") to get all the methods for the print generic.

Value

No return value, called for printing model fit

Author(s)

Siwei Zhand and Bingshu E. Chen

See Also

The default method for print print.default, lplb

Examples

```
#
#  See examples in lplb and lple
#
```

`print.lple` *print a lple object*

Description

print are used to provide a short summary of lple outputs.

Usage

```
## S3 method for class 'lple'
print(x, ...)
```

Arguments

`x`	the results of a lple fit
`...`	other options used in print()

Details

print.lple is called to print object or summary of object from the lple model `lple`.

The default method, print.default has its own help page. Use methods("print") to get all the methods for the print generic.

Value

No return value, called for printing model fit

Author(s)

Siwei Zhand and Bingshu E. Chen

See Also

The default method for print `print.default`, `lple`

Examples

```
#
# see example in lple
#
```

`survfit.lple` *Compute a Survival Curve from a Local Linear Partial Likelihood Estimate.*

Description

Computes the predicted survival function for a model fitted by (lple).

Usage

```
## S3 method for class 'lple'
## S3 method for class 'lple'
survfit(formula, se.fit=TRUE, conf.int=.95, ...)
```

Arguments

`formula`	a fitted model from (lple) fit
`se.fit`	a logical value indicating whether standard errors shall be computed. Default is TRUE
`conf.int`	The level for a two-sided confidence interval on the survival curve. Default is 0.95
`...`	other arguments to the specific method

Details

survfit.lple is called to compuate baseline survival function from the lple model `lple`.

The default method, survfit has its own help page. Use methods("survfit") to get all the methods for the survfit generic.

Value

survfit.lple returns a list of predicted baseline survival function, cumulative hazard function and residuals.

`surv`	Predicted baseline survival function when beta(w) = 0.
`cumhaz`	Baseline cumulative hazard function, -log(surv).
`hazard`	Baseline hazard function.
`varhaz`	Variance of the baseline hazard.
`residuals`	Martingale residuals of the (lple) model.
`std.err`	Standard error for the cumulative hazard function, if se.fit = TRUE.

See `survfit` for more detail about other output values such as upper, lower, conf.type. Confidence interval is based on log-transformation of survival function.

Author(s)

Bingshu E. Chen

See Also

The default method for survfit `survfit`, #`survfit.lple`

Examples

```
#
# See example in lple
#
```

Index