Package: sleev (via r-universe)

October 11, 2026

Type Package

Title Semiparametric Likelihood Estimation with Errors in Variables

Version 1.2.0

Description Efficient regression analysis under general two-phase sampling, where Phase I includes error-prone data and Phase II contains validated data on a subset.

License GPL (>= 2)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends Rcpp (>= 1.0.7), R (>= 3.5.0)

Imports data.table

LinkingTo Rcpp, RcppArmadillo, RcppEigen

Suggests lme4, MASS, rmarkdown, splines, testthat (>= 3.0.0), R.rsp, tibble, knitr, quarto

LazyData true

Config/testthat/edition 3

URL https://github.com/dragontaoran/sleev

BugReports https://github.com/dragontaoran/sleev/issues

Repository https://dragontaoran.r-universe.dev

Date/Publication 2026-03-05 16:39:45 UTC

RemoteUrl https://github.com/dragontaoran/sleev

RemoteRef HEAD

RemoteSha a99f25dbb6e66cdd05631f06e7319aa9e7dfe168

Contents

coefficients.linear2ph . . . 2
coefficients.logistic2ph . . . 3
cv_linear2ph . . . 3
cv_logistic2ph . . . 5
linear2ph . . . 7
logistic2ph . . . 10
mock.vccc . . . 12
print.linear2ph . . . 13
print.logistic2ph . . . 13
print.summary.linear2ph . . . 14
print.summary.logistic2ph . . . 14
spline2ph . . . 15
summary.linear2ph . . . 16
summary.logistic2ph . . . 16

Index **18**

`coefficients.linear2ph` *Extract Coefficients from linear2ph Model*

Description

Extracts estimated coefficients from a two-phase linear regression model of class `linear2ph`.

Usage

```
## S3 method for class 'linear2ph'
coefficients(object, ...)
```

Arguments

`object`	An object of class `linear2ph`.
`...`	Additional arguments passed to other methods.

Value

A numeric vector of coefficients if the model converged, otherwise `NULL` with a warning.

See Also

`coefficients`

coefficients.logistic2ph

Extract Coefficients from logistic2ph Model

Description

Extracts estimated coefficients from a two-phase logistic regression model of class `logistic2ph`.

Usage

```
## S3 method for class 'logistic2ph'
coefficients(object, ...)
```

Arguments

`object`	An object of class `logistic2ph`.
`...`	Additional arguments passed to other methods.

Value

A numeric vector of coefficients if the model converged, otherwise `NULL` with a warning.

See Also

`coefficients`

cv_linear2ph

Cross-validation log-likelihood prediction for `linear2ph`

Description

Performs cross-validation to calculate the average predicted log likelihood for the `linear2ph` function. This function can be used to select the B-spline basis that yields the largest average predicted log likelihood. See pacakge vigenette for code examples.

Usage

```
cv_linear2ph(
  y_unval = NULL,
  y = NULL,
  x_unval = NULL,
  x = NULL,
  z = NULL,
  data = NULL,
  nfolds = 5,
```

```
  max_iter = 2000,
  tol = 1e-04,
  verbose = FALSE
)
```

Arguments

y_unval	Specifies the column of the error-prone outcome that is continuous. Subjects with missing values of y_unval are omitted from the analysis. This argument is required.
y	Specifies the column that stores the validated value of y_unval in the second phase. Subjects with missing values of y are considered as those not selected in the second phase. This argument is required.
x_unval	Specifies the columns of the error-prone covariates. Subjects with missing values of x_unval are omitted from the analysis. This argument is required.
x	Specifies the columns that store the validated values of x_unval in the second phase. Subjects with missing values of x are considered as those not selected in the second phase. This argument is required.
z	Specifies the columns of the accurately measured covariates. Subjects with missing values of z are omitted from the analysis. This argument is optional.
data	Specifies the name of the dataset. This argument is required.
nfolds	Specifies the number of cross-validation folds. The default value is 5. Although nfolds can be as large as the sample size (leave-one-out cross-validation), it is not recommended for large datasets. The smallest value allowable is 3.
max_iter	Specifies the maximum number of iterations in the EM algorithm. The default number is 2000. This argument is optional.
tol	Specifies the convergence criterion in the EM algorithm. The default value is 1E-4. This argument is optional.
verbose	If TRUE, then show details of the analysis. The default value is FALSE.

Details

cv_linear2ph gives log-likelihood prediction for models and data like those in linear2ph. Therefore, the arguments of cv_linear2ph is analogous to that of linear2ph.

Value

cv_linear2ph() returns a list that includes the following components:

avg_pred_loglike	The average predicted log likelihood across each fold.
pred_loglike	The predicted log likelihood in each fold.
converge	The convergence status of the EM algorithm in each run.

Examples

```
## Not run:
  data("mock.vccc")
  # different B-spline sizes
  sns <- c(15, 20, 25, 30, 35, 40)
  # vector to hold mean log-likelihood
  pred_loglike.1 <- rep(NA, length(sns))
  # specify number of folds in the cross validation
  k <- 5
  for (i in 1:length(sns)) {
    # constructing B-spline basis using the same process as in Section 4.3.1
    sn <- sns[i]
    data.sieve <- spline2ph(x = "VL_unval", data = mock.vccc, size = sn,
                            degree = 3, group = "Sex")

    # cross validation, produce mean log-likelihood
    start.time <- Sys.time()
    res.1 <- cv_linear2ph(y = "CD4_val", y_unval = "CD4_unval",
                          x ="VL_val", x_unval = "VL_unval", z = "Sex",
                          data = data.sieve, nfolds = k, max_iter = 2000,
                          tol = 1e-04, verbose = FALSE)
    # save mean log-likelihood result
    pred_loglike.1[i] <- res.1$avg_pred_loglik
  }
  # Print predicted log-likelihood for different B-spline sizes
  print(pred_loglike.1)

## End(Not run)
```

`cv_logistic2ph` *Cross-validation log-likelihood prediction for* `logistic2ph`

Description

Performs cross-validation to calculate the average predicted log likelihood for the `logistic2ph` function. This function can be used to select the B-spline basis that yields the largest average predicted log likelihood. See pacakge vigenette for code examples.

Usage

```
cv_logistic2ph(
  y_unval = NULL,
  y = NULL,
  x_unval = NULL,
  x = NULL,
  z = NULL,
  data,
  nfolds = 5,
```

```
  tol = 1e-04,
  max_iter = 1000,
  verbose = FALSE
)
```

Arguments

y_unval	Column name of the error-prone or unvalidated binary outcome. This argument is optional. If y_unval = NULL (the default), y is treated as error-free.
y	Column name that stores the validated value of y_unval in the second phase. Subjects with missing values of y are considered as those not selected in the second phase. This argument is required.
x_unval	Specifies the columns of the error-prone covariates. This argument is required.
x	Specifies the columns that store the validated values of x_unval in the second phase. Subjects with missing values of x are considered as those not selected in the second phase. This argument is required.
z	Specifies the columns of the accurately measured covariates. Subjects with missing values of z are omitted from the analysis. This argument is optional.
data	Specifies the name of the dataset. This argument is required.
nfolds	Specifies the number of cross-validation folds. The default value is 5. Although nfolds can be as large as the sample size (leave-one-out cross-validation), it is not recommended for large datasets. The smallest value allowable is 3.
tol	Specifies the convergence criterion in the EM algorithm. The default value is 1E-4. This argument is optional.
max_iter	Specifies the maximum number of iterations in the EM algorithm. The default number is 2000. This argument is optional.
verbose	If TRUE, then show details of the analysis. The default value is FALSE.

Details

cv_logistic2ph gives log-likelihood prediction for models and data like those in logistic2ph. Therefore, the arguments of cv_logistic2ph is analogous to that of logistic2ph.

Value

cv_logistic2ph() returns a list that includes the following components:

avg_pred_loglike	Stores the average predicted log likelihood.
pred_loglike	Stores the predicted log likelihood in each fold.
converge	Stores the convergence status of the EM algorithm in each run.

Examples

```
## Not run:
data("mock.vccc")
# different B-spline sizes
sns <- c(15, 20, 25, 30, 35, 40)
# vector to hold mean log-likelihood
pred_loglike.1 <- rep(NA, length(sns))
# specify number of folds in the cross validation
k <- 5
for (i in 1:length(sns)) {
  # constructing B-spline basis using the same process as in Section 4.3.1
  sn <- sns[i]
  data.sieve <- spline2ph(x = "CD4_unval", size = 20, degree = 3,
                          data = mock.vccc, group = "Prior_ART",
                          split_group = TRUE)
  # cross validation, produce mean log-likelihood
  start.time <- Sys.time()
  res.1 <- cv_logistic2ph(y = "ADE_val", y_unval = "ADE_unval",
                          x = "CD4_val", x_unval = "CD4_unval",
                          z = "Prior_ART", data = data.sieve,
                          tol = 1e-04, max_iter = 1000, verbose = FALSE)

  # save mean log-likelihood result
  pred_loglike.1[i] <- res.1$avg_pred_loglik
}
# Print predicted log-likelihood for different B-spline sizes
print(pred_loglike.1)


## End(Not run)
```

`linear2ph` *Sieve maximum likelihood estimator (SMLE) for two-phase linear regression problems*

Description

Performs efficient semiparametric estimation for general two-phase measurement error models when there are errors in both the outcome and covariates. See pacakge vigenette for code examples.

Usage

```
linear2ph(
  y_unval = NULL,
  y = NULL,
  x_unval = NULL,
  x = NULL,
  z = NULL,
```

```
  data = NULL,
  hn_scale = 1,
  se = TRUE,
  tol = 1e-04,
  max_iter = 1000,
  verbose = FALSE
)
```

Arguments

y_unval	Column name of the error-prone or unvalidated continuous outcome. Subjects with missing values of y_unval are omitted from the analysis. This argument is required.
y	Column name that stores the validated value of y_unval in the second phase. Subjects with missing values of y are considered as those not selected in the second phase. This argument is required.
x_unval	Specifies the columns of the error-prone covariates. Subjects with missing values of x_unval are omitted from the analysis. This argument is required.
x	Specifies the columns that store the validated values of x_unval in the second phase. Subjects with missing values of x are considered as those not selected in the second phase. This argument is required.
z	Specifies the columns of the accurately measured covariates. Subjects with missing values of z are omitted from the analysis. This argument is optional.
data	Specifies the name of the dataset. This argument is required.
hn_scale	Specifies the scale of the perturbation constant in the variance estimation. For example, if hn_scale = 0.5, then the perturbation constant is $0.5n^{-1/2}$, where n is the first-phase sample size. The default value is 1. This argument is optional.
se	If FALSE, then the variances of the parameter estimators will not be estimated. The default value is TRUE. This argument is optional.
tol	Specifies the convergence criterion in the EM algorithm. The default value is 1E-4. This argument is optional.
max_iter	Maximum number of iterations in the EM algorithm. The default number is 1000. This argument is optional.
verbose	If TRUE, then show details of the analysis. The default value is FALSE.

Details

Models for linear2ph() are specified through the arguments. The dataset input should at least contain columns for unvalidated error-prone outcome, validated error-prone outcome, unvalidated error-prone covariate(s), validated error-prone covariate(s), and B-spline basis. B-spline basis can be generated from splines::bs() function, with argument x being the unvalidated error-prone covariate(s). See vignette for options in tuning the B-spline basis.

Value

`linear2ph()` returns an object of class `"linear2ph"`. The function `coef()` is used to obtain the coefficients of the fitted model. The function `summary()` is used to obtain and print a summary of results.

An object of class `"linear2ph"` is a list containing at least the following components:

`call`	the matched call.
`coefficients`	A named vector of the linear regression coefficient estimates.
`sigma`	The residual standard error.
`covariance`	The covariance matrix of the linear regression coefficient estimates.
`converge`	In parameter estimation, if the EM algorithm converges, then `converge = TRUE`. Otherwise, `converge = FALSE`.
`converge_cov`	In variance estimation, if the EM algorithm converges, then `converge_cov = TRUE`. Otherwise, `converge_cov = FALSE`.

References

Tao, R., Mercaldo, N. D., Haneuse, S., Maronge, J. M., Rathouz, P. J., Heagerty, P. J., & Schildcrout, J. S. (2021). Two-wave two-phase outcome-dependent sampling designs, with applications to longitudinal binary data. *Statistics in Medicine, 40*(8), 1863–1876. https://doi.org/10.1002/sim.8876

See Also

`cv_linear2ph()` to calculate the average predicted log likelihood of this function.

Examples

```
## Not run:
# Regression model: CD4 ~ VL + Sex. CD4 and VL are partially validated.
data("mock.vccc")
sn <- 20
data.linear <- spline2ph(x = "VL_unval", data = mock.vccc, size = sn,
                         degree = 3,  group = "Sex")
res_linear <- linear2ph(y_unval = "CD4_unval", y = "CD4_val",
                        x_unval = "VL_unval", x = "VL_val",
                        z = "Sex", data = data.linear,  hn_scale = 1,
                        se = TRUE, tol = 1e-04, max_iter = 1000,
                        verbose = FALSE)

## End(Not run)
```

`logistic2ph` *Sieve maximum likelihood estimator (SMLE) for two-phase logistic regression problems*

Description

This function returns the sieve maximum likelihood estimators (SMLE) for the logistic regression model from Lotspeich et al. (2021). See pacakge vigenette for code examples.

Usage

```
logistic2ph(
  y_unval = NULL,
  y = NULL,
  x_unval = NULL,
  x = NULL,
  z = NULL,
  data = NULL,
  hn_scale = 1,
  se = TRUE,
  tol = 1e-04,
  max_iter = 1000,
  verbose = FALSE
)
```

Arguments

`y_unval`	Column name of the error-prone or unvalidated binary outcome. This argument is optional. If `y_unval = NULL` (the default), `y` is treated as error-free.
`y`	Column name that stores the validated value of `y_unval` in the second phase. Subjects with missing values of `y` are considered as those not selected in the second phase. This argument is required.
`x_unval`	Specifies the columns of the error-prone covariates. This argument is required.
`x`	Specifies the columns that store the validated values of `x_unval` in the second phase. Subjects with missing values of `x` are considered as those not selected in the second phase. This argument is required.
`z`	Specifies the columns of the accurately measured covariates. This argument is optional.
`data`	Specifies the name of the dataset. This argument is required.
`hn_scale`	Specifies the scale of the perturbation constant in the variance estimation. For example, if `hn_scale = 0.5`, then the perturbation constant is $0.5n^{-1/2}$, where n is the first-phase sample size. The default value is `1`. This argument is optional.
`se`	If `FALSE`, then the variances of the parameter estimators will not be estimated. The default value is `TRUE`. This argument is optional.

tol	Specifies the convergence criterion in the EM algorithm. The default value is 1E-4. This argument is optional.
max_iter	Maximum number of iterations in the EM algorithm. The default number is 1000. This argument is optional.
verbose	If TRUE, then print messages of the analysis. The default value is FALSE.

Details

Models for logistic2ph() are specified through the arguments. The dataset input should at least contain columns for unvalidated error-prone outcome, validated error-prone outcome, unvalidated error-prone covariate(s), validated error-prone covariate(s), and B-spline basis. B-spline basis can be generated from splines::bs() function, with argument x being the unvalidated error-prone covariate(s). See vignette for options in tuning the B-spline basis.

Value

logistic2ph() returns an object of class "logistic2ph". The function coef() is used to obtain the coefficients of the fitted model. The function summary() is used to obtain and print a summary of results.

An object of class "logistic2ph" is a list containing at least the following components:

call	the matched call.
coefficients	A named vector of the logistic regression coefficient estimates.
covariance	The covariance matrix of the logistic regression coefficient estimates.
converge	In parameter estimation, if the EM algorithm converges, then converge = TRUE. Otherwise, converge = FALSE.
converge_cov	In variance estimation, if the EM algorithm converges, then converge_cov = TRUE. Otherwise, converge_cov = FALSE.

References

Lotspeich, S. C., Shepherd, B. E., Amorim, G. G. C., Shaw, P. A., & Tao, R. (2021). Efficient odds ratio estimation under two-phase sampling using error-prone data from a multi-national HIV research cohort. *Biometrics, biom.13512.* https://doi.org/10.1111/biom.13512

Examples

```
## Not run:
# Regression model: ADE ~ CD4 + Prior_ART. ADE and CD4 are partially validated.
data("mock.vccc")
sn <- 20
data.logistic <- spline2ph(x = "CD4_unval", size = 20, degree = 3,
                           data = mock.vccc, group = "Prior_ART",
                           split_group = TRUE)
res_logistic <- logistic2ph(y = "ADE_val", y_unval = "ADE_unval",
                           x = "CD4_val", x_unval = "CD4_unval",
                            z = "Prior_ART", data = data.logistic,
                            hn_scale = 1/2, se = TRUE, tol = 1e-04,
```

```
                        max_iter = 1000, verbose = FALSE)

## End(Not run)
```

`mock.vccc` *Mock VCCC dataset.*

Description

A simulated dataset constructed to imitate the Vanderbilt Comprehensive Care Clinic (VCCC) patient records, which have been fully validated and therefore contain validated and unvalidated versions of all variables. The VCCC cohort is a good candidate for the purpose of illustration. The data presented in this section are a mocked-up version of the actual data due to confidentiality, but the data structure and features, such as mean and variability, closely resemble the real dataset.

Usage

```
mock.vccc
```

Format

A data frame with 2087 rows and 8 variables:

ID patient ID

VL_unval viral load at antiretroviral therapy (ART) initiation, error-prone outcome, continuous

VL_val viral load at antiretroviral therapy (ART) initiation, validated outcome, continuous

ADE_unval having an AIDS-defining event (ADE) within one year of ART initiation, error-prone outcome, binary

ADE_val having an AIDS-defining event (ADE) within one year of ART initiation, validated outcome, binary

CD4_unval CD4 count at ART initiation, error-prone covariate, continuous

CD4_val CD4 count at ART initiation, validated covariate, continuous

Prior_ART whether patient is ART naive at enrollment, error-free covariate, binary

Sex sex of patient, 1 indicates male and 0 indicates female & error-free covariate, binary

Age age of patient, error-free covariate, continuous

Source

https://www.vanderbilthealth.com/clinic/comprehensive-care-clinic

print.linear2ph *Print Method for linear2ph Objects*

Description

Prints the details of a linear2ph object.

Usage

```
## S3 method for class 'linear2ph'
print(x, ...)
```

Arguments

x	An object of class linear2ph.
...	Additional arguments passed to methods

print.logistic2ph *Print Method for logistic2ph Objects*

Description

Prints the details of a logistic2ph object.

Usage

```
## S3 method for class 'logistic2ph'
print(x, ...)
```

Arguments

x	An object of class logistic2ph.
...	Additional arguments passed to methods

`print.summary.linear2ph` *Print Method for summary.linear2ph Objects*

Description

Prints a structured summary of a `linear2ph` model.

Usage

```
## S3 method for class 'summary.linear2ph'
print(x, ...)
```

Arguments

`x`	An object of class `summary.linear2ph`.
`...`	Additional arguments passed to methods

Value

Invisibly returns `x`.

`print.summary.logistic2ph` *Print Method for summary.logistic2ph Objects*

Description

Prints a structured summary of a `logistic2ph` model.

Usage

```
## S3 method for class 'summary.logistic2ph'
print(x, ...)
```

Arguments

`x`	An object of class `summary.logistic2ph`.
`...`	Additional arguments passed to methods

Value

Invisibly returns `x`.

spline2ph *Splines for two-phase regression functions*

Description

Creates splines for two-phase regression function in this package, including `linear2ph`, `logistic2ph`, `cv_linear2ph`, `cv_logistic2ph`.

Usage

```
spline2ph(
  x,
  data,
  size = 20,
  degree = 3,
  bs_names = NULL,
  group = NULL,
  split_group = TRUE
)
```

Arguments

`x`	Column names of the covariate of the dataset.
`data`	Specifies the name of the dataset. This argument is required.
`size`	Pass on to the `df` argument in `splines::bs()`. Degrees of freedom for EACH variable.
`degree`	Pass on to the `degree` argument in `splines::bs()`. Degree of the piecewise polynomial. Default is 3 for cubic splines.
`bs_names`	Optional. Vecotr of column names of the output B-spline basis matrix. When not specified, a default will be provided.
`group`	Optional. Column name of the categorical variable of which might have heterogeneous errors among different groups.
`split_group`	Optional. Whether to split by group proportion for the group with B-spline size if the `group` argument is provided. If `FALSE`, then the split will be averaged across all groups. Default is `TRUE`.

Details

This function can be directly applied for regression model with one or more error-prone continuous covariates.

Value

the `data.frame` object including the original dataset and the B-spline bases.

Examples

```
# example code
data("mock.vccc")
sn <- 20
data.linear <- spline2ph(x = "VL_unval", data = mock.vccc, size = sn,
                         degree = 3,  group = "Sex")
```

`summary.linear2ph` *Summary Method for linear2ph Objects*

Description

Summarizes the details of a `linear2ph` object.

Usage

```
## S3 method for class 'linear2ph'
summary(object, ...)
```

Arguments

`object`	An object of class `linear2ph`.
`...`	Additional arguments passed to methods

Value

An object of class `summary.linear2ph`, containing the call, coefficients, and covariance.

`summary.logistic2ph` *Summary Method for logistic2ph Objects*

Description

Summarizes the details of a `logistic2ph` object.

Usage

```
## S3 method for class 'logistic2ph'
summary(object, ...)
```

Arguments

`object`	An object of class `logistic2ph`.
`...`	Additional arguments passed to methods

Value

An object of class `summary.logistic2ph`, containing the call, coefficients, and covariance.

Index

∗ **datasets**
mock.vccc, 12

coef.linear2ph
(coefficients.linear2ph), 2
coef.logistic2ph
(coefficients.logistic2ph), 3
coefficients, *2*, *3*
coefficients.linear2ph, 2
coefficients.logistic2ph, 3
cv_linear2ph, 3
cv_linear2ph(), *9*
cv_logistic2ph, 5

linear2ph, 7
logistic2ph, 10

mock.vccc, 12

print.linear2ph, 13
print.logistic2ph, 13
print.summary.linear2ph, 14
print.summary.logistic2ph, 14

spline2ph, 15
summary.linear2ph, 16
summary.logistic2ph, 16