Package 'aftsem'

September 15, 2024

Type Package

Title Semiparametric Accelerated Failure Time Model

Version 1.0

Date 2024-09-01

Maintainer Martin Benedikt <benedma2@cvut.cz>

Description Implements several basic algorithms for estimating regression parameters for semiparametric accelerated failure time (AFT) model. The main methods are: Jin rank-based method (Jin (2003) <doi:10.1093/biomet/90.2.341>), Heller's estimating method (Heller (2012) <doi:10.1198/016214506000001257>), Polynomial smoothed Gehan function method (Chung (2013) <doi:10.1007/s11222-012-9333-9>), Buckley-James method (Buckley (1979) <doi:10.2307/2335161>) and Jin`s improved least squares method (Jin (2006) <doi:10.1093/biomet/93.1.147>). This package can be used for modeling right-censored data and for comparing different estimation algorithms.

License GPL (>= 3)

BugReports https://github.com/benedma2/aftsem-package/issues

Imports survival, Rcpp (>= 1.0.10), stats, quantreg, optimx

URL https://github.com/benedma2/aftsem-package

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.1

Depends R (>= 4.2.0)

Encoding UTF-8

NeedsCompilation yes

Author Martin Benedikt [aut, cre]

Repository CRAN

Date/Publication 2024-09-15 20:30:21 UTC

Contents

`aftsem-package` *Semiparametric Accelerated Failure Time Model*

Description

Implements several basic algorithms for estimating regression parameters for semiparametric accelerated failure time (AFT) model. The main methods are: Jin rank-based method (Jin (2003) <doi:10.1093/biomet/90.2.341>), Heller's estimating method (Heller (2012) <doi:10.1198/016214506000001257>), Polynomial smoothed Gehan function method (Chung (2013) <doi:10.1007/s11222-012-9333-9>), Buckley-James method (Buckley (1979) <doi:10.2307/2335161>) and Jin's improved least squares method (Jin (2006) <doi:10.1093/biomet/93.1.147>). This package can be used for modeling right-censored data and for comparing different estimation algorithms.

Details

The DESCRIPTION file:

Package:	aftsem
Type:	Package
Title:	Semiparametric Accelerated Failure Time Model
Version:	1.0
Date:	2024-09-01
Maintainer:	Martin Benedikt <benedma2@cvut.cz>
Description:	Implements several basic algorithms for estimating regression parameters for semiparametric accelerated fail
License:	GPL (>= 3)
BugReports:	https://github.com/benedma2/aftsem-package/issues
Imports:	survival, Rcpp (>= 1.0.10), stats, quantreg, optimx
URL:	https://github.com/benedma2/aftsem-package
Authors@R:	person(given = "Martin", family = "Benedikt",role = c("aut", "cre"),email = "benedma2@cvut.cz")
LinkingTo:	Rcpp, RcppArmadillo
RoxygenNote:	7.3.1
LazyData:	true
Depends:	R (>= 4.2.0)

Encoding:	UTF-8
Author:	Martin Benedikt [aut, cre]

Index of help topics:

aftsem	Accelerated Failure Time Semiparametric Model
aftsem-package	Semiparametric Accelerated Failure Time Model
aftsem.control	Control list for package
aftsem_fit	Semi-parametric AFT Model Fitting
gehan_estimation	Gehan's Estimation for Survival Data
gehan_heller_estimation	Gehan-Heller Estimation of regression parameters
gehan_poly_estimation	Estimation of Regression Parameters from Smoothed Gehan Function
print.aftsem	Print method for aftsem xs
print.summaryaftsem	Print method for objects of class 'summaryaftsem'
summary.aftsem	Summary function for aftsem package

Author(s)

Martin Benedikt [aut, cre]

Maintainer: Martin Benedikt <benedma2@cvut.cz>

References

Buckley, J.; James, I. Linear Regression with Censored Data. Biometrika. 1979, issn 00063444.

Jin, Z., Lin, D.Y., Wei, L. J., and Ying, Z. (2003). Rank-based inference for the accelerated failure time models, Biometrika, 90, 341-353.

Heller, G. Smoothed rank regression with censored data. Journal of the American Statistical Association. 2007

See Also

survfit aftgee

`aftsem` *Accelerated Failure Time Semiparametric Model*

Description

Accelerated Failure Time Semiparametric Model

Usage

```
aftsem(
  formula,
  data,
  control = aftsem.control(),
  method = "buckley",
  binit = "auto",
  ties = NULL,
  na.action = na.omit,
  subset = NULL,
  resample = 0,
  ...
)
```

Arguments

formula	A formula expression, of the form response ~ predictors. Response must be a Surv object
data	An optional data.frame in which to interpret the variables in the formula.
control	Control parameters for the AFT model.
method	A character string specifying the method to be used (buckley,jin,gehan,gehan-heller,gehan-poly).
binit	Initial values for the regression parameters.
ties	A method to handle ties in the failure times. If ties = NULL only warning will be printed. If ties = jitter, the data will be augumented
na.action	A method to deal with missing values (na.fail)
subset	An optional vector specifying a subset of observations to be used in the fitting process.
resample	Number of resamples for variance estimation for gehan and jin methods.
...	Additional arguments.

Value

A list representing the fit - ‘call‘: Call of the function - ‘cnames‘: Column names - ‘method‘: Method of estimation - ‘nobs‘: Number of observations - ‘censored‘: Number of censored observations - ‘betafirst‘: Initial beta - ‘epsilon‘: Epsilon in convergence criterion - ‘max_iterations‘: Max iterations for buckley and jin method - ‘resample‘: Resample number - ‘objects from aftsem.fit‘: All the object from fit function

Examples

```
# Generating example data
library(survival)
set.seed(123) # for reproducibility
n <- 100 # number of observations
Z <- matrix(rnorm(n*2), ncol = 2) # two covariates
```

```
beta <- c(0.5, -0.25) # true coefficients
times <- exp(Z %*% beta + rnorm(n)) # simulated survival times
censoring <- runif(n,0,30)
observed_times <- times
delta <- 1 * (times<=censoring)

# Fit the model


fit <- aftsem(Surv(log(observed_times), delta) ~ Z[,1] + Z[,2],
              method = "buckley",
              binit = "auto",
              ties = "NULL",
              na.action = na.omit,
              subset = NULL
)

# Print the summary
summary(fit)
```

`aftsem.control` *Control list for package*

Description

Control list for package

Usage

```
aftsem.control(
  eps = 10^-5,
  maxiter = 15,
  gehan_eps = 10^-6,
  optimx.alg = "BFGS",
  variance.estimation = FALSE,
  quantile.method = "br",
  use.grad = FALSE
)
```

Arguments

`eps`	Convergence criterion
`maxiter`	Maximum iterations for algorithms
`gehan_eps`	Epsilon value for polynomial Gehan optimalization
`optimx.alg`	Algorithm that will be used in optimx minimalization (see optimx documentation for more details)

`variance.estimation`
: If hellers sd will be estimated

`quantile.method`
: Method used for quantile regression minimalization

`use.grad`
: If excact gradient will be used instead of the numerical one, default is numerical == FALSE

Value

list of parameters above

Note

When alternating the control list, one must write other variables also. Example: When user want to estimate the Hellers covariance matrix he would need to change the control list -> aftsem(....., control = list(variance.estimation = TRUE, use.grad = FALSE, optimx.alg = "BFGS))

`aftsem_fit` *Semi-parametric AFT Model Fitting*

Description

Fits a semi-parametric accelerated failure time (AFT) model to the provided data using various methods.

Usage

```
aftsem_fit(Z, y, delta, betafirst, method, control, intercept, resample, nobs)
```

Arguments

`Z`
: A matrix of covariates.

`y`
: A vector of the response variable, typically survival times.

`delta`
: A censoring indicator vector where 1 indicates an uncensored observation and 0 indicates a censored observation.

`betafirst`
: The initial estimate of the beta coefficients.

`method`
: The method of estimation to use, one of "buckley", "gehan", "jin", or "gehan-poly".

`control`
: A list of control parameters including ‘eps‘ for convergence criterion and ‘max-iter‘ for the maximum number of iterations.

`intercept`
: Logical; if TRUE, include an intercept in the model.

`resample`
: The number of resamples to use for Monte Carlo estimation of variance; relevant for certain methods only.

`nobs`
: The number of observations in the data.

Details

The ‘aftsem_fit‘ function provides a way to fit a semi-parametric AFT model to survival data with potential RIGHT censoring. Depending on the chosen method, different estimation techniques are used, such as Buckley-James or Gehan’s method. If resampling is required for the method, the function will generate resamples from an exponential distribution.

Value

Returns a list object of class "aftsem" containing the following components: - ‘converged‘: Logical indicating if the fitting procedure converged. - ‘beta‘: The estimated beta coefficients. - ‘iters‘: The number of iterations performed. - ‘resid‘: The residuals from the model fit. NOT THE MARTINGALE RESIDUALS - ‘sampling.used‘: Logical indicating if sampling was used. - ‘intercept‘: The estimated intercept, included if ‘intercept = TRUE‘. - ‘beta_star‘: The beta coefficients estimated for each resample, included if resampling was used. - ‘fe‘: Number of calls of function in minimalization proccess (only available for gehan-poly and gehan-heller method) - ‘covariance‘ Covariance matrix (only available for gehan-heller method)

---

gehan_estimation *Gehan’s Estimation for Survival Data*

Description

This function performs Gehan’s estimation of regression parameters proposed by Jin

Usage

```
gehan_estimation(y, Z, delta, rsmat, m, init = FALSE)
```

Arguments

y	A numeric vector of survival times.
Z	A matrix of covariates
delta	A numeric vector indicating censoring status
rsmat	A resampling matrix
m	Method for quantreg optimalization
init	A logical value indicating whether to return the initial fit object (default is ‘FALSE‘). If ‘FALSE‘, only the coefficients are returned.

Value

If ‘init = FALSE‘ and ‘change == 1‘, returns a list with elements ‘INTERCEPT‘, ‘RESID‘, ‘ITERS‘, ‘CONVERGED‘, ‘BETA‘. Otherwise, returns a matrix of resampled Gehan estimates.

Note

This function is a slightly different version from the original by Zherzen Jin, part of the now not available ‘lss‘ program.

Author(s)

Zherzen Jin

`gehan_heller_estimation` *Gehan-Heller Estimation of regression parameters*

Description

Gehan-Heller Estimation of regression parameters

Usage

```
gehan_heller_estimation(
  y,
  Z,
  delta,
  binit,
  optimx.alg,
  variance.estimation,
  use.grad
)
```

Arguments

`y`	Numeric vector of survival times or times to event/censoring.
`Z`	Numeric matrix of covariates with observations in rows and covariates in columns.
`delta`	Numeric vector indicating censoring, with 1 for an event and 0 for censored observations.
`binit`	Numeric vector or matrix for initial estimates of regression coefficients.
`optimx.alg`	Optimalization algorithm that will be used (see optimx package documentation for more information)
`variance.estimation`	If covariance matrix will be estimated
`use.grad`	Indicator wheter numerical or excact gradient will be used, default is FALSE == numerical
	Covariance estimation is programmed but not tested!

Value

A list containing the estimated regression coefficients ('BETA'), residuals ('RESID'), and the number of iterations taken by the optimization routine ('ITERS').

Note

The recommend use is with numerical aproximation of gradient. The true gradiet can be sensitive for initial beta values (binit). For Covariance estimation please set the variance.estimation in control list to TRUE.

`gehan_poly_estimation` *Estimation of Regression Parameters from Smoothed Gehan Function*

Description

Estimates regression parameters by optimizing a smoothed version of Gehan's statistic.

Usage

```
gehan_poly_estimation(y, Z, delta, binit, epsilon, optimx.alg, use.grad)
```

Arguments

`y`	A numeric vector of the response variable, survival times.
`Z`	A matrix of covariates.
`delta`	A censoring indicator vector where 1 indicates an uncensored observation and 0 indicates a censored observation.
`binit`	Initial values for the beta coefficients.
`epsilon`	Smoothing parameter.
`optimx.alg`	Optimalization algorithm that will be used (see optimx package documentation for more information)
`use.grad`	Indicator wheter numerical or excact gradient will be used, default is FALSE == numerical

Details

The 'gehan_poly_estimation' function performs estimation of regression parameters by minimizing the smoothed Gehan's loss function.

Value

A list containing: - 'BETA': The estimated beta coefficients. - 'RESID': The residuals from the model fit. - 'ITERS': The number of iterations performed during optimization.

print.aftsem *Print method for aftsem xs*

Description

Prints a summary of an aftsem model fit x.

Usage

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

Arguments

x	An x of class "aftsem", typically the result of a call to ‘aftsem_fit‘.
...	Further arguments passed to or from other methods.

Details

The ‘print.aftsem‘ method provides a user-friendly summary of the model fit, including the method used for parameter estimation, convergence status, estimated parameters, number of iterations, and the percentage of censored observations.

Value

The function is called for its side effect, which is printing the summary to the console. It invisibly returns NULL.

See Also

aftsem_fit for model fitting.

print.summaryaftsem *Print method for objects of class ‘summaryaftsem‘*

Description

Print method for objects of class ‘summaryaftsem‘

Usage

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

Arguments

x	An object of class ‘summaryaftsem‘
...	Further arguments passed to or from other methods.

Value

The function prints object ‘summaryaftsem‘

`summary.aftsem` *Summary function for aftsem package*

Description

Provides a summary of an aftsem model fit, including the model call, residuals, initial and final coefficient estimates, method, convergence status, number of iterations, number of observations, percent of censored observations, and if available, the estimated covariance matrix of the coefficients, standard deviations, z-values, and p-values for a Wald test.

Usage

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

Arguments

object	An object of aftsemfit
...	Further arguments passed to or from other methods.

Value

An object of class ‘summaryaftsem‘ that contains summary information of the fitted aftsem model.

Index